AMHERST SYSTEMS INC.

2

SIMULATOR NETWORKING PHASE I

Technical Report – Study/Services

February 1, 1992 Through October 2, 1992

2 October 1992

Document Control Number: 611-9160002

Contract No.: N61339-92-C-0036

CDRL Sequence No.: A003

Classification: UNCLASSIFIED

Prepared for:

Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3275

Prepared by:

Amherst Systems Inc.
30 Wilson Road
Buffalo, NY 14221-7026

AMHERST SYSTEMS INC.

SIMULATOR NETWORKING PHASE I

Technical Report – Study/Services

February 1, 1992 Through October 2, 1992

2 October 1992

Document Number: 611-91600002

Prepared for:	Naval Training Systems Center 12350 Research Parkway Orlando, FL 32826-3275	Douglas C. Reif Task Enginee
		Bruce D. Friedman Task Engineer
Contract No.:	N61339-92-C-0036	Douglas C. Reif Program Manager
CDRL Items:	A003	Daniel G. Sentz Quality Assurance

DTIC QUALITY INSPECTED 5

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

AMHERST SYSTEMS INC. • 30 Wilson Road • Buffalo, New York 14221-7026

AMHERST SYSTEMS INC.

The Contractor, Amherst Systems Inc., hereby certifies that, to the best of its knowledge and belief, the technical data delivered herewith under Contract No. N61339-92-C-0036 is complete, accurate, and complies with all requirements of the contract.

10/5/92
Date

Douglas C. Reif
Douglas C. Reif, Program Manager

TABLE OF CONTENTS

SECTION		PAGE

TABLE OF CONTENTS - CONTINUED

TABLE OF CONTENTS - CONTINUED

LIST OF ILLUSTRATIONS

LIST OF TABLES

Section 1

INTRODUCTION

1.1 Identification

This document is a Technical Report - Study/Services, Contract Data Requirement List (CDRL) data item A003 for the Simulator Networking Phase I SBIR Project, contract number N61339-92-C-0036.

Amherst Systems is working to add Distributed Interactive Simulation (DIS) capability to the Integrated Test Bed (ITB) facility aircraft simulator for the Avionics Directorate at Wright Laboratory (WL), Wright-Patterson AFB. The Avionics Directorate is the Air Force research laboratory for advanced avionics concepts. Current research areas include multiprocessor architectures, defensive avionics, and integrated communication, navigation and identification (CNI) systems. The Avionics Directorate is actively working to integrate its facilities so that systems and concepts can be developed, tested, and evaluated in a multispectral environment. The ITB simulator is the vehicle for advanced testing. It is an engineering level simulator that is easily reconfigurable to play the role of many different aircraft. The ITB simulator features fully programmable flight dynamics capability, an out-the-window display, terrain, and avionics displays. Operational systems such as electronic sensors and weapons can be added based on the desired application. The ITB simulator supports man-in-the-loop testing for systems and concepts in software based simulations.

Although DIS capability has been implemented in several aircraft simulators, there has been little rigorous analysis of the ability of DIS to perform aerial combat and air mission support roles. The ITB is highly qualified to support testing in this area. The ITB can simulate F-15E, A-7D, and C-130 aircraft and other airframes could be supported at modest cost. The Flight Dynamics Directorate, also at Wright-Patterson AFB, is in the process of adding DIS capability to some of their experimental aircraft simulators. The Flight Dynamics Directorate simulators are also reconfigurable and used to model advanced aircraft designs. Since the Avionics Directorate and the Flight Dynamics Directorate are closely located, there is a unique opportunity to perform basic research with DIS in a timely and cost-effective manner. Amherst Systems plans to support coordinated testing using DIS with the Avionics Directorate and the Flight Dynamics Directorate. The ITB would then be used in a joint services training exercise in a close air support mission, applying knowledge gained from the tests. Many benefits will be gained from this approach, including:

a. To succeed in modern combat, cooperation among the armed services is vital. Most training efforts to date using DIS have been performed by the Army only. This project demonstrates Air Force commitment to DIS and joint services training efforts.

b. The project will advance basic understanding of the operation of DIS. It is expected that requirements for aerial combat will stress its capability. Important knowledge about its limitations and techniques to avoid them will be gained. Areas that require further study will be noted and suggestions on possible improvements will be made.

c. The advanced capabilities of the simulators used in the tests means that these modern aircraft simulators could be used in training exercises. This provides a cost-effective method of testing the capabilities of different aircraft in realistic roles. Other services will have the opportunity to train with air support representing advanced Air Force capability. This could be especially relevant for special forces training.

d. The reconfigurable capabilities of the laboratories can provide training for specialized missions that cannot be supported by point design simulators.

e. Implementation of DIS in simulators such as the ITB promotes synergism and cooperation between the research and training branches of the armed forces, groups that normally operate independently.

1.2 Summary of Results

This section summarizes the work to date on this SBIR Phase I project.

1.2.1 Work Plan

In order to investigate the feasibility of adding DIS capability to the ITB simulator during this effort, a work plan was developed. The objectives of the work plan were to identify the requirements for adding DIS capability to the ITB and to develop a design that would satisfy these requirements. At this point, it is be possible to determine cost and schedule for implementing the capability. A Phase II proposal has been submitted to the program sponsors.

The adopted work plan is as follows:

a. Study the ITB to understand its functionality.

b. Study the DIS PDU's and understand their usage.

c. Derive system requirements for adding DIS capability to the ITB.

d. Evaluate the data items in each transmitted PDU and identify their source in the ITB modules. Evaluate the data items in each received PDU and identify their destination and use in the ITB modules.

e. Develop interface requirements, both for exchanging data among ITB modules and for connecting the ITB to other DIS sites.

f. Produce a top-level design. The modules and data structures affected by the enhancement of the ITB to include DIS capability will be identified. Pseudo-code examples are given for some important functions (for example, see Figure 2.3.2.1.2-2).

Items (a), (b) and (c) were completed by April 30, 1992. A complete discussion of these items is contained in Appendix A of the document "Simulator Networking Phase I, Contractor's Progress, Status and Management Plan, February 1, 1992 Through April 30, 1992", previously submitted during this program. Section 2.1 presents the results of this earlier work. The other items are discussed in detail in the remainder of this report.

1.2.2 Results of Analytic Efforts

The objectives of the work plan have been achieved in this Phase I project. A Phase II proposal has been submitted to the program sponsors in which the tasks, cost, and schedule for implementing DIS capability in the ITB has been justified. A summary of the results of the tasks from the work plan is given below.

a. Study the ITB. Amherst Systems received extensive documentation about the ITB from WL/AAAS-2. The documents were studied, and an understanding of the architecture and functional organization of the ITB was gained. Mr. John Woodyard, ITB technical liaison at Wright Laboratory was extremely helpful in providing documentation and answering questions about ITB operation. Refer to Appendix A.40 of the interim report for detailed information about the ITB.

b. Study the DIS PDU's. Each of the DIS PDU's was studied. The fields of each PDU were examined and the types of data for each were identified. Of particular importance were the coordinate systems used in most of the PDU's, and the dead reckoning information. The Institute for Simulation and Training (IST) was consulted on several occasions and was very helpful. Appendix A.50 of the interim report presents this information.

c. Derive system requirements for adding DIS capability to the ITB. Before this task could be addressed, a mission and configuration for the ITB had to be decided in order to focus the work for a realistic Phase II project. It was determined that the close air support mission would be an appropriate role. Since most DIS training applications to date have been ground battles, the ability to provide robust close air support would greatly enhance the realism and scope of the training. The following ITB configuration is considered appropriate for this mission:

- F-15E Airframe
- Out-the-Window (OTW) Display
- Forward Looking Infrared (FLIR) Sensor/Display
- Head's-Up Display (HUD)
- Laser Ranger Sensor
- MK-82 Weapon System
- Basic Navigation Avionics

The requirements for implementing this capability were developed. It was determined that the Entity State PDU was the most important, and that the Fire PDU, Detonation PDU and Collision PDU will also be implemented. The Repair/Resupply PDU's will be implemented only in rudimentary form since the ITB will not have this capability. The experimental PDU's will not be implemented in Phase II. The requirements to support the selected PDU's, the ITB functions, and connectivity to other DIS sites, were developed. Important areas addressed include gaming area, participants, the message interface, coordinate transformation, dead reckoning, and time. The system requirements are discussed in detail in Appendix A.60 of the interim report.

d. Evaluate PDU data items. The selected DIS PDU's were re-examined. The fields of each were analyzed to determine their source of information in the ITB (for transmitted PDU's), or used and destination (for received PDU's). Section 2.2 of this report discusses the PDU data items.

e. Develop interface requirements. External interface requirements were developed for linking the ITB to other DIS sites and sources of data. The external connections include a GPS receiver module for Universal Coordinated Time, the Automated Forces Testbed for testing, and other DIS sites for training missions. The data interchange within the ITB was examined. New requirements to support DIS capability affected the types of data exchanged among ITB tasks. New data structures to hold DIS related information were developed and the types of data elaborated. Performance requirements for ITB software were derived. The interface and performance requirements are presented in Section 2.3.

f. Produce top-level design. New tasks to support DIS capability have been designed. The interrelationships between existing ITB tasks and the new tasks have been developed. This report elaborates on the functionality of the new tasks as well as the enhancements that must be made to the existing tasks. Perhaps the most important technical challenge for Phase II is to reconcile the WGS-1984 ECEF coordinates used in DIS with the coordinate systems used in the ITB and its BBN OTW/FLIR displays. This is critically important for proper scenario representation. A detailed discussion of the issues related to coordinate transformation is provided in Section 2.4.

1.3 Conclusions

Perhaps the most important conclusion derived from this study is that implementation of DIS capability in the ITB simulator is feasible. The analytic work performed in Phase I substantiates this conclusion. The requirements for adding DIS capability and a design that implements these requirements have been developed. What is significant about this work is that the ITB is easily reconfigurable to simulate many types of aircraft and their operational systems. The ITB is an ideal simulator to support research in this area as well as participate in training programs that require advanced Air Force capability.

Section 2
ANALYTIC RESULTS

2.1 Previous Results

This section summarizes the results of analysis performed in the February 1 to April 30 time period. These results are fully documented in "Simulator Networking Phase I, Contractor's Progress, Status, and Management Plan, February 1, 1992 Through April 30, 1992.

2.1.1 Introduction to the Integrated Test Bed (ITB) Facility

The ITB facility tests and evaluates advanced avionics architectures and equipment for laboratory experimentation requiring a full mission simulation. The facility includes a realistic crewstation mock-up with the elements of digital avionics operating as they would in a real military aircraft. Out-the-Window (OTW) scene generation is also provided, giving the pilot a symbolic background scene outside the crewstation for visual orientation necessary to locate items such as targets, runways, and waypoints. Key ITB support equipment includes the facility processors and systems and their resident software, which provide the test control, monitoring, and simulation necessary to perform avionics system evaluations and demonstrations. The following are the important facility processors and systems:

a. Simulation Host Processor Complex: The simulation software resides on three Harris host processors with shared memory. It simulates a selected aircraft, its environment, sensors, and weapons subsystems, and performs data collection and evaluation. The simulation provides real-time inputs to the crewstation and to mission software via the Multiplex Bus Terminal (MBT) and MIL-STD-1553B multiplex data bus. The Harris host processors are currently being upgraded to an Alliant computer system.

b. Generalized Avionics Simulation/Integration System (GENASIS): The GENASIS is a crewstation mock-up employing two processors to provide realistic pilot interfaces through integrated displays, command entry keyboards, backup instruments, and aircraft controls (i.e., stick, rudder, and throttle). Resident software on the MicroVAX processor provides control of input and output data from the Harris complex to the crewstation via Direct Memory Access (DMA). Resident software on the MicroVAX III processor provides communication with the MIL-STD-1533B data bus via the Bus Interface Unit (BIU).

c. OTW Scene Generation Complex: The BBN display system, video control system, and their respective resident software provide various visual background scenes along with a simulated Head-Up Display (HUD).

The Model Simulation System (MSS) provides the simulation of avionic sensors, weapon subsystems, and an airframe/environment for a military aircraft performing a Close Air Support (CAS) mission. Figure 2.1.1-1 show , an overview of the MSS. The MSS employs the Availability Demonstration Simulation Support (ADSS) Executive as its simulation executive.

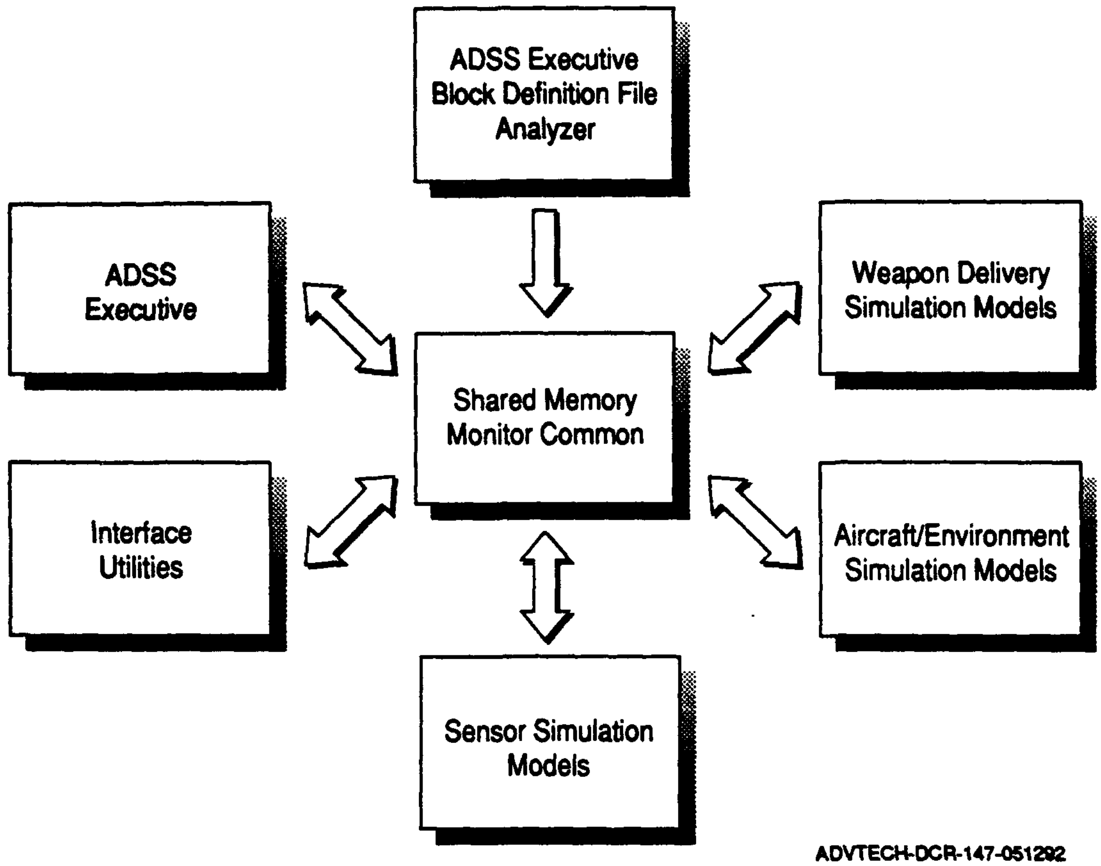

Figure 2.1.1-1 Overview of MSS Software

The MSS consists of the five following functional types of software:

a. ADSS Executive: The ADSS Executive controls the simulation, interfaces the simulation with bus input/output (I/O), and provides checkpoint/restart capability to reset the simulation to a previous time in the scenario.

b. Sensor Simulation Models: The Sensor Simulation Models simulate the on-board aircraft sensor systems.

c. Weapon Delivery Simulation Models: These Models simulate functional responses of weapon systems to the operational stores management software.

d. Aircraft/Environment Simulation Models: The Aircraft/Environment Models provide airframe dynamics, flight control dynamics, and propulsion information.

e. Interface Utilities: The Interface Utilities provide access to the crewstation cockpit controls and displays, and the OTW generation complex.

In addition, the MSS includes the Scenario Generator/Acceptance Test Program. This software replaces the aircraft environment models by providing idealized navigation, guidance and aircraft dynamics for MSS testing.

2.1.2 System Requirements Summary

The system requirements for implementing DIS capability in the ITB were discussed in detail in the document "Simulator Networking Phase I, Contractor's Progress, Status and Management Plan". As a result, this section will summarize the findings.

a. Connect the ITB to a long-haul network. The Integrated Test Bed Facility in Building 620 at Wright-Patterson AFB is not currently connected to any long-haul network that can support DIS. There is a T-1 line connection on base to the Air Force Institute of Technology (AFIT). A fiber optic link exists between Building 620 and AFIT, but a dedicated connection must be made to unite the resources. Figure 2.1.2-1 illustrates required connectivity among Wright-Patterson AFB facilities, much of which is either planned or already in place.

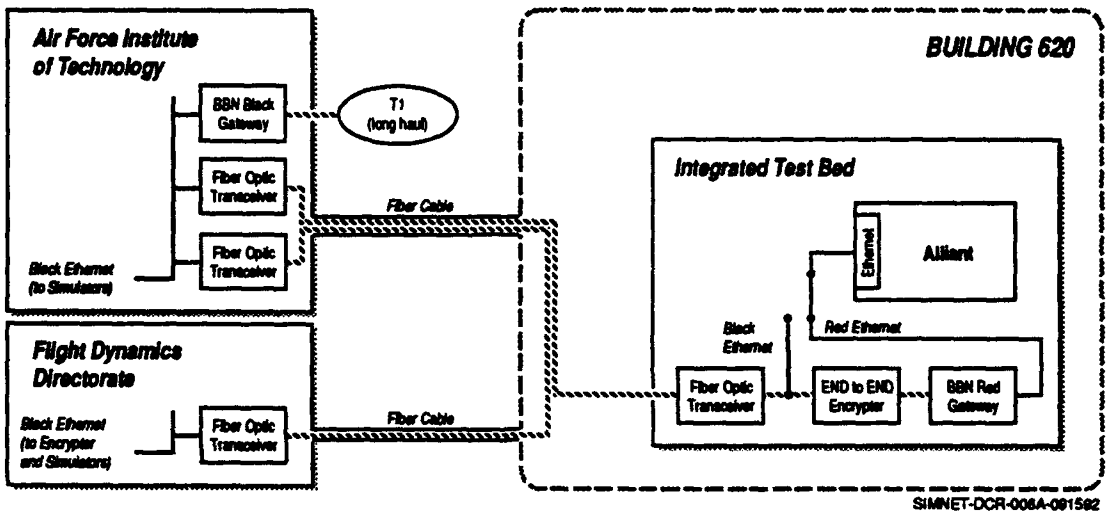

Figure 2.1.2-1 Connectivity Among Wright-Patterson AFB Facilities

b. Selectability of operation. The operator should be able to select between DIS and non-DIS operation. The ITB must continue to be able to support Wright Laboratory projects after it has been augmented with DIS capability.

c. Gaming area. The gaming area must be selectable by the operator. This will be coordinated with other DIS participants before scenario execution begins. Terrain data corresponding to the selected gaming area must be used. Concerns that must be addressed related to gaming selection are listed below. Many of these concerns represent general simulation problems. Project 2851 is currently addressing these items.

- Terrain and Underlying Earth Models. The ITB software, the aircraft avionics, and the OTW display all use independent algorithms to model the earth and its terrain. These models must be reconciled.

- Gaming Area Effects: Surface characteristics affect sensor responses. For example, flying through mountainous terrain produces multipath effects in RF sensors. The systems models used in the ITB must be able to account for terrain effects.

- Cultural/Environmental Interactions: Cultural and environmental features affect detectability in obvious ways. For instance, a tank convoy that would be visible at great distance in the desert could be virtually invisible in a forested area. The visual displays and sensor models must account for this.

- Commonality of Simulators: This a concern with all DIS applications. Even if common environment, terrain and cultural databases are used, differences in interpretation can invalidate scenario results. Typical problems involve line-of-sight, camouflage/appearance and resolution.

d. Participants. The ITB must be able to realistically represent all participants in the scenario. This is critically important for all sensing functions, including the out-the-window display. If visual confirmation is required before engagement, the ITB must provide sufficient detail to perform this. There are several aspects to entity presentation. The first is a realistic silhouette, which is often provided with wire-frame diagrams. The second is static contrast/reflectivity/emissitivity. Tanks should be realistically colored, and aircraft should present accurate radar cross-sections. The third is that the articulated parts need to be properly represented, if required. From a sensing point of view, the ITB should be able to produce realistic cross-sections in all spectra for all participants/aspect angles as well as for cultural and environmental features.

e. General Conduct. Procedures must be established for executing DIS scenarios. These procedures include establishing the type of player and mission the ITB will represent, procuring terrain/environmental databases, checking out the communication link, coordinating with the "allied commander" and other players, configuring the ITB, and starting execution at the prescribed time.

f. Sending and Receiving DIS Messages. The ITB must be capable of sending and receiving messages using DIS protocol data units. It is helpful to look at this capability as being performed at three levels: physical, logical and virtual.

  - Physical: There must be interface hardware to establish the link. The physical link includes all the cabling and processor boards required to transmit and receive the PDU's.

  - Logical: The logical link represents the software drivers and control processes that must be developed to control the physical link. The logical link must operate with minimal delay. It is anticipated that a driver will exist for the interface hardware, but that additional software will be required to provide efficient processing within the ITB.

  - Virtual: The virtual link allows the ITB software to operate with read/write PDU functionality. This operates at OSI level 7 and is independent of the underlying hardware interface.

g. Message Generation. The ITB must be capable of generating PDU's for each function that it can perform. Table 2.1.2-1 maps functionality to PDU generation capability.

Table 2.1.2-1 PDU Generation/Capability Cross Reference

Capability	PDU
Always required	Entity State, Collision
Weapon firing	Fire, Detonation
Resupply	Resupply Offer, Resupply Cancel
Receive supplies	Service Request, Resupply Received, Resupply Cancel
Repair	Repair Complete
Receive repairs	Service Request, Repair Response

f. Message Reception. The ITB shall be capable of receiving PDU's of any type. Undefined PDU's (PDU Type other than 1 through 10) will be ignored. Table 2.1.2-2 summarizes the actions that will be taken in response to the defined PDU's.

Table 2.1.2-2 Actions for Received PDU's

Received PDU	Action
Entity State	Create the entity if it has not been seen before. Use information from the entity type database to determine what to do with the data from the PDU. Update visual and sensor processing and displays. Reset dead reckoning for the entity.
Fire	Represent muzzle flash effects if necessary.
Detonation	Determine damage, visual, sensor effects on ITB.
Collision	If collision is with ITB and ITB has comparably determined that a collision has occurred, calculate effects on ITB.
Service Request Resupply Offer Resupply Received Resupply Cancel Repair Complete Repair Response	The ITB will have no repair/resupply capability. If a Service Request PDU is received, the ITB will respond that it is unable to provide supplies or repairs. Receipt of the other PDU's will be ignored.

i. Coordinate Transformations. The DIS standard for entity location is earth-centered earth-fixed (ECEF) coordinates based on the centroid of the WGS-1984 earth using the IEEE 754-1985 standard for floating point numbers, in units of meters. The ITB is currently being upgraded from spherical to WGS-1984 earth, and software is being transferred from Harris to Alliant computers. The ITB documentation has not yet been modified to reflect the changes. What is most important is that each of the software components and hardware display units use equivalent earth interpretations. The affected components are the ITB flight software, munition trajectory calculations, the OTW/FLIR display and the HUD. The following transformations must be performed:

- ITB aircraft location/orientation to ECEF/entity
- Munition detonation location to ECEF/entity

- ECEF/entity to OTW/FLIR location/orientation

- ITB aircraft location/orientation to OTW/FLIR location/orientation

- ITB aircraft location/orientation to HUD location/orientation

It may be necessary to derive transformations to reconcile different coordinate systems. For example, to convert from WGS-1984 to spherical earth requires knowing the radius of the spherical earth to ensure that the entity is in the same location in both coordinate systems. Velocity and orientation may also require transformation since, in WGS-1984 earth, unlike spherical earth, one degree of latitude is not a constant distance and down does not always point to the center of the earth. Unless these conditions are properly handled, realistic simulation cannot occur.

j. Dead Reckoning of ITB Managed Entities. Dead reckoning is performed on ITB managed entities to determine when to issue Entity State PDU's. Each Entity State PDU contains dead reckoning information to allow other sites to update the location and orientation of the entity between PDU's. Before simulation begins, each entity is assigned three threshold values - location, orientation and time. An entity must issue and Entity State PDU if one of the following situations occurs. The situations apply both to the basic entity and any of its articulated parts. It should be noted that the Entity State PDU can be issued for other reasons also (e.g. change in appearance).

- Its actual location differs from its dead reckoned location by more than the threshold.

- Its actual orientation differs form its dead reckoned orientation by more than the threshold.

- The elapsed time since the most recent Entity State PDU was issued exceeds the time threshold.

To implement the thresholding, the DIS interface software must perform dead reckoning calculations on the aircraft at the rate at which it is updated (64 Hz) and compare results to the actual values computed by the ITB operational software. When a threshold is exceeded, the software will generate an Entity State PDU and reset dead reckoning calculations with the new actual values. One important concern is whether the 64 Hz update rate will cause excessive PDU generation during high G tactical maneuvers. This will be an area of study during the SBIR Phase II program.

k. Dead Reckoning of Other Entities. Dead reckoning is used on other entities to determine their location and orientation at times between their Entity State PDU's. The Entity State PDU's contain the information required to perform the dead reckoning calculations. This computed information will be stored in the shared memory common and provides data to support the sensor and visual displays. The dead reckoning calculations will be performed at a 15 Hz rate, to match the OTW and FLIR updates.

l. Time. The DIS standard permits use of either Universal Coordinated Time or local simulator (relative) time. The ITB presently supports only relative time. Access to Universal Coordinated Time requires reception of GPS satellite signals. WL/AAAS-2 is actively investigating whether the ITB can gain access to this time standard. No matter which time standard is used, tight synchronization among simulator sites is a critical requirement. For coordinated aerial maneuvers, such as welded-wing, errors of milliseconds could cause a collision. The effects of time errors will be a subject of serious study during Phase II.

2.2 Analysis of PDU's

DIS entities interact with each other through an exchange of messages also known as protocol data units (PDU's): The DIS standard consists of ten PDU's, as listed in Table 2.2-1.

Table 2.2-1 Standard DIS PDU's

Value	PDU Kind	Description
1	Entity State	Information about an entity
2	Fire	Firing of a weapon
3	Detonation	Detonation or impact of munition
4	Collision	Collisions between entities
5	Service Request	Request for logistics
6	Resupply Offer	Offering of supplies
7	Resupply Received	Receipt of supplies
8	Resupply Cancel	Cancellation of service request
9	Repair Complete	Completion of repairs
10	Repair Response	Acknowledgment of repair completion

The Entity State, Fire, Detonation, and Collision PDU's are important to ITB operation. The Entity State PDU, which represents motion and appearance of the simulated aircraft, is most critical. The Repair/Resupply PDU's (values 5 to 10) will be supported only to the extent that the ITB will respond correctly, denying the capability, should other entities request these services. It should be noted here that a number of temporary/experimental PDU's are included in the DIS standard. Some of them, particularly the emissions related PDU's, are strongly related to ITB operation. However, since these PDU's are still undergoing testing and have not been formally approved, no analysis on them has been performed to date.

Analysis of the PDU's has been categorized as transmitted and received. The processing requirements for the transmission and receipt of PDU's are quite different. Being DIS-compliant means being able to transmit PDU's as defined in the draft standard. A early key goal in the Phase II implementation will make the ITB DIS-compliant. But this alone does not guarantee that the ITB can be a meaningful participant in training scenarios. There are open questions as to whether the DIS protocol can realistically support a fast mover such as the ITB in aerial combat and close air support missions. Additionally, for the ITB to perform realistically, the received PDU's must be processed in a timely manner and their data seamlessly injected into the ITB programs and routines.

The DIS interface has functionality independent of that of the ITB cockpit. In other words, the ITB operation should appear the same whether entities are generated in an internal scenario or created externally through DIS. The act of sending and receiving PDU's is an obvious DIS interface function in that it has no direct effect on the ITB activity. As another example, the DIS protocol states that an Entity

State PDU must be generated whenever the dead reckoned position/orientation and actual position/orientation vary by more than a certain amount. This is quite independent of the aero models used to simulate ITB motion. In fact, a special process dedicated to monitoring the variation will probably be required. Yet another example is the coordinate/unit/representation transformations to and from ITB and DIS variables. Many of the DIS functions will be supported in independently executing tasks running in parallel to the baseline ITB tasks. Section 2.4 describes the approach in further detail.

2.2.1 Transmitted PDU's

The Entity State PDU is the most critical to the early stages of testing close air support scenarios. If the opposing forces are to act in a realistic manner (e.g., taking cover, firing defensively), the appearance of the ITB must be represented in the opposition forces simulators with high fidelity. The actual representation within those simulators is beyond the scope of this project. However, the ITB will be responsible for generating accurate and timely Entity State PDU's. When the ITB is able to accomplish this, open questions about fidelity can be addressed. The critical issue is whether ITB activity (particularly movement) can be represented realistically in other simulators. Some of the key variables in the analysis are as follows:

a. Location and orientation thresholds for PDU generation.
b. Communication and processing latencies.
c. Terrain and feature representation/correlation (particularly important for low level flights).

Some of the important observations that will be included in the analysis are as follows:

a. Location/orientation errors over time.
b. "Jumping" of position as PDU's are received.
c. Amount of message traffic.

Another consideration in ITB modeling is the use of articulated parts. The reason for modeling the articulated parts is that other players can detect and react to their settings. Conversely, if they cannot be detected and they have no effect, there is no need to model them. The wing stations could be represented, if for no other reason than realistic visual presentation in high resolution simulators. The ITB internal software area model also represents aircraft components such as flaps and the speed brake. Articulated parts on the ITB will not be implemented in Phase II. The Phase II Final Report will discuss whether articulated parts would have improved functionality and also analyze how they could be implemented.

The ITB could potentially model more than the aircraft itself. The ITB software supports the modeling of Maverick missile firing and flight, although this capability is not yet fully implemented. If the missile were detectable by other players, it would need to be represented as an independent entity in DIS. Maverick missile capability will not be implemented during Phase II. However, the DIS interface software will be developed in such a manner that additional entities could be supported at a future date.

The Collision PDU is important in low-level flight and when aircraft fly in formation. Representation of terrain and cultural/environmental features take on great significance when low-level flying inherent in close air support missions is considered. Collision detection will be implemented by placing an imaginary shell around each entity. If the shells overlap, a collision will have occurred.

Analysis of the transmitted PDU's is summarized in the following tables. Source/value columns in the tables annotated "n/a" are not applicable to the Phase II work but may be added in future implementations. Source/value columns annotated "**" means that these values are assigned before the exercise and will remain constant during the entire period of the exercise. Report Section and Appendix references in the tables are for the DIS-standards document "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation." Table 2.2.1-1 provides a cross-reference between PDU figure definitions in the DIS-standards document and the tables in this section.

Table 2.2.1-1 Transmit PDU/Table Cross-Reference

PDU	DIS-Standards Figure	Table Number
Entity State	5-19	2.2.1-2
Fire	5-20	2.2.1-3
Detonation	5-21	2.2.1-4
Collision	5-28	2.2.1-5
Resupply Cancel	5-25	2.2.1-6
Repair Complete	5-26	2.2.1-7

Table 2.2.1-2 Transmit PDU: Entity State

Field	Source	Activity/Notes	Source/Value
Protocol Version	Fixed	Assigned before exercise	**
Exercise Identifier	Fixed	Assigned before exercise (4.7.1.2)	**
PDU Type	Fixed	Constant 1	1
Entity Identifier		Assigned by DIS interface for ITB cockpit and each other entity created by ITB software.	
Site Identifier	Fixed	Assigned by authority managing DIS exercises	**
Host Identifier	Fixed	Assigned by authority managing DIS exercises. One site may have several hosts.	**
Entity Identifier	Fixed at entity creation	Each entity created by host has its own unique entity identifier.	1
Force Identifier	Fixed at entity creation	Enumerated type, Appendix E.30.1.1	1
Entity Type		DIS interface must know about types of entities that ITB can create.	
Entity Kind	Fixed at entity creation	Enumerated type, Appendix H1.30.1	1
Domain	Fixed at entity creation	Enumerated type, value depends on entity kind, Appendix H1.30.1 subsections	2
Country	Fixed at entity creation	Enumerated type, zero for environmental and cultural feature entity kind, Appendix F.30.1.2	168
Category	Fixed at entity creation	Enumerated type Platform: Appendix H2.30.2 Munition: Appendix H2.30.3 Life Form: Appendix H1.30.1.3.3 Environmental: Appendix H1.30.1.4.3 Cultural: Appendix H1.30.1.5.3	1
Subcategory	Fixed at entity creation, possibly modified due to attrition	Enumerated type, (a.k.a.) Platform: (Designation) H2.30.2 Munition: (Designation) H2.30.3 Life Form: (Designation) H1.30.1.3.3 Environmental: (Size) H1.30.1.4.3 Cultural: (Size) H1.30.1.5.3	7
Specific	Fixed at entity creation, possibly modified due to attrition	Enumerated type, (a.k.a.) Platform: (Model) H2.30.2 Munition: (Model) H2.30.3 Life Form: (Number) H1.30.1.3.3 Environmental: (Unused) H1.30.1.4.3 Cultural: (Unused) H1.30.1.5.3	n/a
Extra	Fixed at entity creation	Not used for any entity kind	n/a

Table 2.2.1-2 Transmit PDU: Entity State - (Continued)

Alternate Entity Type Entity Kind Domain Country Category Subcategory Specific Extra	Fixed at entity creation, possible modified due to attrition	Same formats as Entity Type, purpose is that some players may have alternative view of the entity. Its use will depend on the requirements of others in the simulation.	n/a
Time Stamp	Universal Coordinated Time (UTC) or internal simulator (relative time)	Indicates the time at which the data is valid. Up to one hour can be represented, units are $3600/2^{31}$ seconds. Bit zero is 0 if relative, 1 if UTC. See 5.2.19. Represents time at which ITB data is valid.	UTC will be used if it is installed.
Entity Location X-component Y-component Z-component	Variable, based on location of a particular part of the entity, represents the origin of the entity coordinate system	Geocentric Cartesian Coordinates (World Coordinates) origin is the centroid of the earth, positive x-axis passes through Prime Meridian and Equator, positive y-axis through E90° and Equator, positive z-axis through North Pole. Each component is 64-bit floating point, conforming to IEEE 754-1985. The shape of earth is based on World Geodetic System 1984 survey (WGS-1984). Units are meters. Based on data taken from ITB database.	Derived from ACLATR, ACLNGR, ACALTF in program EARTH
Entity Linear Velocity X-component Y-component Z-component	Variable, based on instantaneous velocity at time of PDU	Vector relative to axes used to represent the Entity Location (World Coordinates). Each component is a 32-bit floating point, conforming to IEEE 754-1985. Units are meters per second. Based on data taken from ITB database.	Derived from X1DOTF, Y1DOTF, Z1DOTF, in program AIRPLN

Table 2.2.1-2 Transmit PDU: Entity State - (Continued)

Field	Source	Activity/Notes	Source/Value
Entity Orientation Psi Theta Phi	Variable, based on instantaneous orientation at time of PDU	Represents transformation from Geocentric Cartesian Coordinates to Entity Coordinates. Entity Coordinates origin at, the center of the bounding volume, independent of articulated parts. Positive x-axis is front, positive y-axis is right, positive z-axis is down. (Psi, Theta, Phi) represent standard Euler angles. Units are called Binary Angle Measurement (BAM) = $360/2^{32}$ degrees. Based on data taken from ITB database.	Derived from THETAR, ACPHIR, ACPSIR in program AIRPLN
Dead Reckoning Parameters		Parameters for ITB and other created entities to be determined by DIS interface. Algorithm to be based on entity type.	
Dead Reckoning Algorithm	Fixed at entity creation	Enumerated type, Appendix I.30.7	4 [DRM (R,V,W)]
Linear Acceleration X-component Y-component Z-component	Variable, based on instantaneous acceleration at time of PDU	Vector in Entity Coordinates. Each component is a 32-bit floating point number, conforming to IEEE 754-1985. Units are meters per second per second.	Derived from UDOTTF, VDOTTF, WDOTTF in program AIRPLN
Angular velocity Rate about x-axis Rate about y-axis Rate about z-axis	Variable, based on instantaneous angular velocity at time of PDU	Rate at which entity's orientation is changing. Each rate is a 32-bit integer in units of BAM's per millisecond, relative to Entity Coordinates. The positive direction is counter-clockwise about each axis.	Derived from PBODYR, QBODYR, RBODYR in program AIRPLN
Other parameters	TBD	TBD	n/a
Entity Appearance	Variable, depending on state of entity at time of PDU.	Bit field definitions dependent on Entity Kind and Domain, defined in Appendix D.	Use of this field will be determined in Phase II.
Entity Marking		Special markings on ITB aircraft.	
Character set	Fixed at entity creation	Appendix F.30.1.4	1
1st Character ... 11th Character	Fixed at entity creation, possible modified.	String of characters, can be interpreted for display. Most significant (first) character at start of string. Unused characters set to blank.	**
Capabilities	Fixed at entity creation	Bit field definitions, defined in 5.2.7	0
Number of Articulation Parameters	Variable, depending on state of entity at time of PDU	Number of Articulation Parameter records to follow.	0
Articulation Parameters	Repeated, actual number may vary	DIS interface must get information from ITB. DIS interface will perform dead reckoning calculations associated with articulated parts.	
Change indicator	Variable	Set to one at start, incremented with each change in the articulated part	n/a

Table 2.2.1-2 Transmit PDU: Entity State - (Continued)

Field	Source	Activity/Notes	Source/Value
ID-attached to	Fixed at entity creation	Part articulation number index of entity/part to which it is attached. Zero is main entity, one or higher indicates attachment to other articulated part.	n/a
Parameter type	Fixed at entity creation	Definitions in Appendix G, quite complicated. Some types indicate part definition and enumeration, other part definition, enumeration and type of data to follow. The second variation may have several parameters types to fully define it.	n/a
Parameter value	Variable, depending on state of entity at time of PDU	Dependent on parameter type, 64 bit field. May be a numeric value, enumeration value or an entity type. Numeric units and format depend on the type of value represented. See Appendix G.	n/a

Table 2.2.1-3 Transmit PDU: Fire

Field	Source	Activity/Notes	Source/Value
Protocol Version	Fixed	Assigned before exercise	**
Exercise Identifier	Fixed	Assigned before exercise (4.7.1.2)	**
PDU Type	Fixed	Constant 2	2
Firing Entity ID Site Host Entity	Fixed	See Entity Identifier for Entity State PDU	Same as Entity State PDU
Target Entity ID Site Host Entity	Fixed at time of PDU	All zeroes if intended targets is unknown. See Entity Identifier for Entity State PDU	All zeroes
Munition ID Site Host Entity	Fixed at time of PDU	All zeros mean munition not represented as separate entity and therefore cannot be detected or tracked. Whether the munition is modeled explicitly depends on the sensing capabilities of the other players in the scenario. This should be determined before the start. An explicitly modeled munition requires its own munition model in the ITB software. See Entity Identifier for Entity State PDU.	All zeroes
Event ID		Assigned by DIS interface.	
Site Identifier	Fixed	See "Site Identifier" from Entity Identifier for Entity State PDU	**
Host Identifier	Fixed	See "Host Identifier" from Entity Identifier for Entity State PDU	**
Event Identifier	Variable	Used to link Fire PDU to Detonation PDU. Event sequences use a common event identifier. The event identifier starts at one and is incremented at the start of each sequence.	A sequential counter of releases will be added to routine WEAPEX.
Time Stamp	UTC or internal simulator time	See Time Stamp for Entity State PDU	UTC will be used if it is installed.
Location in World X-component Y-component Z-component	Variable, based on location from which munitions were launched	Location in world coordinates. See Entity Location for Entity State PDU. It is an offset from the world location of the aircraft.	Fixed offset to current aircraft position, offset is TBD.
Burst Descriptor		Other simulators use this to produce firing effects. This data will be extracted from the weapon model.	
Munition	Fixed at time of PDU	Entity Type record with enumerated kind set to munition. This is required whether or not the munition is represented as a separate entity.	Bombs are not defined in the standard.

Table 2.2.1-3 Transmit PDU: Fire - (Continued)

Field	Source	Activity/Notes	Source/Value
Warhead	Fixed at time of PDU	Enumerated type, Appendix B.30.1	Selectable by ITB operator at start of scenario.
Fuze	Fixed at time of PDU	Enumerated type, Appendix B.30.1	Selectable by ITB operator at start of scenario
Quantity	Fixed at time of PDU	Number of rounds fired in this burst, 16-bit unsigned integer.	See Note 1 below
Rate	Fixed at time of PDU	Rounds per minute for the specified munition. If quantity is one, rate will be set to zero, 16-bit unsigned integer.	See Note 1 below
Velocity X-component Y-component Z-component	Instantaneous velocity of munition at time of PDU	Linear velocity of munition at time of firing in world coordinated, 32-bit floating point conforming to IEEE 754-1985, in units of meters per second. It is derived from the velocity of the aircraft.	Same as aircraft.
Range	Fixed at time of PDU	Range in meters that entity's fire control system assumes in computing fire control solution. Set to zero if unknown or unavailable 32-bit floating point, IEEE 754-1985, in units of meters. This is extracted from the weapon model.	From trajectory calculations in routine TRAJEC.

Note 1: The MK-82 release is programmable for number of bombs to release and their expected spacing upon impact. This is dependent on the aircraft altitude, orientation, and velocity at the time of release. The quantity and rate will be calculated based from these values.

Table 2.2.1-4 Transmit PDU: Detonation

Field	Source	Activity/Notes	Source/Value
Protocol Version	Fixed	Assigned before exercise	**
Exercise Identifier	Fixed	Assigned before exercise (4.7.1.2)	**
PDU Type	Fixed	Constant 3	3
Firing Entity ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	Same as Fire PDU
Target Entity ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	All zeroes
Munition ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	All zeroes
Event ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	Same as Fire PDU
Time Stamp	UTC or internal simulator time of detonation	See Time Stamp for Entity State PDU The weapon model must return the time of detonation.	Calculated by routine TRAJEC. UTC will be used if it is installed.
Location in World X-component Y-component Z-component	Variable, based on location where detonation occurs	Location in world coordinates. See Entity Location for Entity State PDU. This information must come from the weapon model.	Calculated by routine TRAJEC.
Burst Descriptor Munition Warhead Fuze Quantity Rate	Fixed at time of Fire PDU	Same as corresponding Fire PDU.	Same as Fire PDU
Velocity X-component Y-component Z-component	Instantaneous velocity of munition immediately prior to detonation	Linear velocity of munition in world coordinates, 32-bit floating point, conforming to IEEE 754-1985, in units of meters per second. This information must come from the weapon model.	Calculated by routine TRAJEC.
Location in Entity Coordinates X-component Y-component Z-component	Variable, depending on target location	Location of detonation with respect to the target, in the target's entity coordinates. This field will be zero if no Target Entity ID is specified. This is determined by the DIS interface.	All zeroes

Detonation Result	Determined at time of PDU	Enumerated type, Appendix F.30.1.3, from weapon model. 0=Detonate, 1=Impact.	Bomb must be properly armed by pilot to detonate.
Number of Articulation Parameters	TBD	Number of Articulation Parameter records to follow, not clear whether weapon model or DIS interface would determine this.	0
Articulation Parameters Change indicator ID-attached to Parameter type Parameter value	TBD	See Articulation Parameters for Entity State PDU.	n/a

Table 2.2.1-5 Transmit PDU: Collision

Field	Source	Activity/Notes	Source/Value
Protocol Version	Fixed	Assigned before exercise	**
Exercise Identifier	Fixed	Assigned before exercise (4.7.1.2)	**
PDU Type	Fixed	Constant 4	4
Issuing Entity ID Site Host Entity	Fixed	See Event Identifier for Entity State PDU	Same as Entity State PDU
Colliding Entity ID Site Host Entity	Determined by collision detection algorithm	Entity Identifier of object that the issuer collided with, zero if unknown or terrain. If a Collision PDU is also received from the colliding entity or collision is with terrain, damage to ITB must be computed.	Calculated by ITB Collision Evaluation task.
Event ID Site Host Entity	Fixed at time of PDU	See Entity ID of Fire PDU	A sequential counter of collisions will be maintained by the ITB Collision Evaluation task.
Time Stamp	UTC or internal simulation time of collision	See Time Stamp for Entity State PDU	UTC will be used if it is installed.
Velocity X-component Y-component Z-component	Instantaneous velocity of issuer at time of collision	Linear velocity of issuer at time of collision in world coordinates, 32-bit floating point, conforming to IEEE 754-1985, in units of meters per second	Aircraft velocity at time of impact.
Mass	Instantaneous mass of issuer at time of collision	Mass of issuing entity, 32-bit floating point, conforming to IEEE 754-1985, in units of kilograms	Aircraft mass at time of impact, includes fuel, external stores, etc.
Location with Respect to Entity X-component Y-component Z-component	Variable, depending on target location	Location of collision with respect to the colliding entity in the colliding entity's coordinates. This field will be zero if no colliding Entity ID is specified.	Calculated by ITB Collision Evaluation task.

Table 2.2.1-6 Transmit PDU: Resupply Cancel

Field	Source	Activity/Notes	Source/Value
Protocol Version	Fixed	Assigned before exercise	**
Exercise Identifier	Fixed	Assigned before exercise (4.7.1.2)	**
PDU Type	Fixed	Constant 8	8
Requesting Entity ID Site Host Entity	Fixed at time of Service Request PDU	Same as corresponding Service Request PDU	Entity ID verbatim from Service Request PDU.
Servicing Entity ID Site Host Entity	Fixed	See Entity Identifier for Entity State PDU	Entity ID of ITB.

Table 2.2.1-7 Transmit PDU: Repair Complete

Field	Source	Activity/Notes	Source/Value
Protocol Version	Fixed at start	Assigned before exercise	**
Exercise Identifier	Fixed	Assigned before exercise (4.7.1.2)	**
PDU Type	Fixed	Constant 9	9
Requesting Entity ID Site Host Entity	Fixed at time of Service Request PDU	Same as corresponding Service Request PDU	Entity ID verbatim from Service Request PDU.
Servicing Entity ID Site Host Entity	Fixed	See Entity Identifier for Entity State PDU	Entity ID of ITB.
Repair	Fixed	No repairs performed	0

2.2.2 Received PDU's

The received PDU's have direct effect over the fidelity of the simulation as viewed by the pilot of the ITB cockpit. The critical concerns involve taking the information from the PDU's and injecting it into the functional components of the ITB software (sensors models, out-the-window display and so on) in a timely manner. Some new capabilities, such as dead reckoning calculations, will need to be added. Important observational tests will be performed during Phase II when the ability to receive PDU's has been implemented. There have been SIMNET problems concerning discrete platform jumps associated with entity state data. This may affect displays and sensor models. It could have a great effect on coordinated platform movement (such as formations) as well as aerial combat and missile evasion.

Analysis of the received PDU's is summarized in the following tables. Report Section and Appendix references in the tables are for the DIS-standards document "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation. Table 2.2.2-1 provides a cross-reference between PDU figure definitions in the DIS-standards document and the tables in this section.

Table 2.2.2-1 Receive PDU Figure/Table Cross-Reference

PDU	DIS-Standards Figure	Table Number
Entity State	5-19	2.2.2-2
Fire	5-20	2.2.2-3
Detonation	5-21	2.2.2-4
Collision	5-28	2.2.2-5
Service Request	5-22	2.2.2-6

Table 2.2.2-2 Receive PDU: Entity State

Field	Source	Activity/Notes	Use/Destination
Protocol Version	Fixed	Must match ITB capability.	Error if no match
Exercise Identifier	Fixed	Must be as expected.	Error if no match
PDU Type	Fixed	Constant 1	
Entity Identifier		Fields uniquely identify each entity. Table of entities maintained in DIS interface. Must be able to translate to created ITB entries for sensor, weapon, display functions.	Create new entry in Entity Data Structures if first appearance.
Site Identifier	Fixed	Assigned by authority managing DIS exercises	May affect time stamp interpretation
Host Identifier	Fixed	Assigned by authority managing DIS exercises. One site may have several hosts.	May affect time stamp interpretation
Entity Identifier	Fixed at entity creation	Each entity created by host has its own unique entity identifier.	
Force Identifier	Fixed at entity creation	Enumerated type, Appendix E.30.1.1, informative.	Informational
Entity Type		Defines appearance of entity to sensors, OTW display, etc. This information must be injected into the ITB. DIS interface must have knowledge of ITB categorization of entities.	Used to access Entity Database, database entries accessed by key.
Entity Kind	Fixed at entity creation	Enumerated type, Appendix H1.30.1	Part of key
Domain	Fixed at entity creation	Enumerated type, value depends on entity kind, Appendix H1.30.1 subsections	May aid visual representation
Country	Fixed at entity creation	Enumerated type, zero for environmental and cultural feature entity kind, Appendix F.30.1.2	Part of key
Category	Fixed at entity creation	Enumerated type Platform: Appendix H2.30.2 Munition: Appendix H2.30.3 Life Form: Appendix H1.30.1.3.3 Environmental: Appendix H1.30.1.4.3 Cultural: Appendix H1.30.1.5.3	Part of key
Subcategory	Fixed at entity creation, possibly modified due to attrition	Enumerated type, (a.k.a.) Platform: (Designation) H2.30.2 Munition: (Designation) H2.30.3 Life Form: (Designation) H1.30.1.3.3 Environmental: (Size) H1.30.1.4.3 Cultural: (Size) H1.30.1.5.3	Part of key
Specific	Fixed at entity creation, possibly modified due to attrition	Enumerated type, (a.k.a.) Platform: (Model) H2.30.2 Munition: (Model) H2.30.3 Life Form: (Number) H1.30.1.3.3 Environmental: (Unused) H1.30.1.4.3 Cultural: (Unused) H1.30.1.5.3	Part of key
Extra	Fixed at entity creation	Not used for any entity kind	

Table 2.2.2-2 Receive PDU: Entity State - (Continued)

Field	Source	Activity/Notes	Use/Destination
Alternate Entity Type Entity Kind Domain Country Category Subcategory Specific Extra	Fixed at entity creation, possible modified due to attrition	Same formats as Entity Type, purpose is that some players may have alternative view of the entity. Its use will depend on the requirements of others in the simulation.	Will not be used.
Time Stamp	Universal Coordinated Time (UTC) or internal simulator (relative time)	Indicates the time at which the data is valid. Up to one hour can be represented, units are $3600/2^{31}$ seconds. Bit zero is 0 if relative, 1 if UTC. See 5.2.19. DIS interface will update deal reckoned location from this time to current time.	Used in dead reckoning calculations.
Entity Location X-component Y-component Z-component	Variable, based on location of a particular part of the entity, represents the origin of the entity coordinate system	Geocentric Cartesian Coordinates (World Coordinates) origin is the centroid of the earth, positive x-axis passes through Prime Meridian and Equator, positive y-axis through E90° and Equator, positive z-axis through North Pole. Each component is 64-bit floating point, conforming to IEEE 754-1985. The shape of earth is based on World Geodetic System 1984 survey (WGS-1984). Units are meters. Must be translated into units accepted by ITB.	Used in dead reckoning calculations.
Entity Linear Velocity X-component Y-component Z-component	Variable, based on instantaneous velocity at time of PDU	Vector relative to axes used to represent the Entity Location (World Coordinates). Each component is a 32-bit floating point, conforming to IEEE 754-1985. Units are meters per second.	Used in dead reckoning calculations.
Entity Orientation Psi Theta Phi	Variable, based on instantaneous orientation at time of PDU	Represents transformation from Geocentric Cartesian Coordinates to Entity Coordinates. Entity Coordinates origin at, the center of the bounding volume, independent of articulated parts. Positive x-axis is front, positive y-axis is right, positive z-axis is down. (Psi, Theta, Phi) represent standard Euler angles. Units are called Binary Angle Measurement (BAM) = $360/2^{32}$ degrees. Must be translated into unites accepted by ITB.	Used in dead reckoning calculations.
Dead Reckoning Parameters		Dead reckoning is performed in the DIS interface. The computed and translated location and orientation are passed to the ITB.	Appendix I.30.6 lists formulas for each dead reckoning algorithm.
Dead Reckoning Algorithm	Fixed at entity creation	Enumerated type, Appendix I.30.7	Determines type of dead reckoning algorithm to use.
Linear Acceleration X-component Y-component Z-component	Variable, based on instantaneous acceleration at time of PDU	Vector in Entity Coordinates. Each component is a 32-bit floating point number, conforming to IEEE 754-1985. Units are meters per second per second.	Used in dead reckoning calculations.

Table 2.2.2-2 Receive PDU: Entity State - (Continued)

Field	Source	Activity/Notes	Use/Destination
Angular velocity Rate about x-axis Rate about y-axis Rate about z-axis	Variable, based on instantaneous angular velocity at time of PDU	Rate at which entity's orientation is changing. Each rate is a 32-bit integer in units of BAM's per millisecond, relative to Entity Coordinates. The positive direction is counter-clockwise about each axis.	Used in dead reckoning calculations.
Other parameters	TBD	TBD	
Entity Appearance	Variable, depending on state of entity at time of PDU.	Bit field definitions dependent on Entity Kind and Domain, defined in Appendix D. Defines appearance of entity to sensors.	May be used in OTW/FLIR displays.
Entity Marking		Defines appearance of entity to sensors.	
Character set	Fixed at entity creation	Appendix F.30.1.4	
1^{st} Character ... 11^{th} Character	Fixed at entity creation, possible modified.	String of characters, can be interpreted for display. Most significant (first) character at start of string. Unused characters set to blank.	Informational, not applicable to initial implementation.
Capabilities	Fixed at entity creation	Bit field definitions, defined in 5.2.7, implicit knowledge of capability of others.	
Number of Articulation Parameters	Variable, depending on state of entity at time of PDU	Number of Articulation Parameter records to follow.	Will not be used in initial implementation.
Articulation Parameters	Repeated, actual number may vary	DIS interface must have knowledge of the representation capabilities of the ITB for each type of entity. Data must be translated to conform to ITB representation.	
Change indicator	Variable	Set to one at start, incremented with each change in the articulated part.	
ID-attached to	Fixed at entity creation	Part articulation number index of entity/part to which it is attached. Zero is main entity, one or higher indicates attachment to other articulated part.	
Parameter type	Fixed at entity creation	Definitions in Appendix G, quite complicated. Some types indicate part definition and enumeration, other part definition, enumeration and type of data to follow. The second variation may have several parameters types to fully define it.	
Parameter value	Variable, depending on state of entity at time of PDU	Dependent on parameter type, 64 bit field. May be a numeric value, enumeration value or an entity type. Numeric units and format depend on the type of value represented. See Appendix G.	

Table 2.2.2-3 Receive PDU: Fire

Field	Source	Activity/Notes	Use/Destination
Protocol Version	Fixed	Must match ITB capability.	Error if no match.
Exercise Identifier	Fixed	Must be as expected.	Error if no match.
PDU Type	Fixed	Constant 2	
Firing Entity ID Site Host Entity	Fixed	See Entity Identifier for Entity State PDU. Can be used to create effects such as muzzle flash.	Error if Firing entity not in Entity Data Structures
Target Entity ID Site Host Entity	Fixed at time of PDU	All zeroes if intended targets is unknown. See Entity Identifier for Entity State PDU.	Informative
Munition ID Site Host Entity	Fixed at time of PDU	All zeros mean munition not represented as separate entity and therefore cannot be detected or tracked. See Entity Identifier for Entity State PDU. Non zero value means new entity for munition must be created.	Create new entity in Entity Data Structures if first appearance.
Event ID		Information maintained in DIS interface.	Save Firing Entity Munition ID.
Site Identifier	Fixed	See "Site Identifier" from Entity Identifier for Entity State PDU.	
Host Identifier	Fixed	See "Host Identifier" from Entity Identifier for Entity State PDU.	
Event Identifier	Variable	Used to link Fire PDU to Detonation PDU. Event sequences use a common event identifier. The event identifier starts at one and is incremented at the start of each sequence.	Remove munition on detonation.
Time Stamp	UTC or internal simulator time	See Time Stamp for Entity State PDU	Informational
Location in World X-component Y-component Z-component	Variable, based on location from which munitions were launched	Location in world coordinates. See Entity Location for Entity State PDU. Represents where munitions were launched, not centroid of firing entity.	Used in dead reckoning. This can be used to locate firing effects.
Burst Descriptor		Used to create firing effects in ITB.	Informative in initial implementation.
Munition	Fixed at time of firing	Entity Type record with enumerated kind set to munition. This is required whether or not the munition is represented as a separate entity.	
Warhead	Fixed at time of firing	Enumerated type, Appendix B.30.1	
Fuze	Fixed at time of firing	Enumerated type, Appendix B.30.1	
Quantity	Fixed at time of firing	Number of rounds fired in this burst, 16-bit unsigned integer.	

Table 2.2.2-3 Receive PDU: Fire - (Continued)

Field	Source	Activity/Notes	Use/Destination
Rate	Fixed at time of firing	Rounds per minute for the specified munition. If quantity is one, rate will be set to zero, 16-bit unsigned integer.	
Velocity X-component Y-component Z-component	Instantaneous velocity of munition at time of PDU	Linear velocity of munition at time of firing in world coordinates, 32-bit floating point conforming to IEEE 754-1985, in units of meters per second.	Used in dead reckoning.
Range	Fixed at time of firing	Range in meters that entity's fire control system assumes in computing fire control solution. Set to zero if unknown or unavailable 32-bit floating point, IEEE 754-1985, in units of meters. This is extracted from the weapon model.	Informative

Table 2.2.2-4 Receive PDU: Detonation

Field	Source	Activity/Notes	Use/Destination
Protocol Version	Fixed	Must match ITB capability.	Error if no match.
Exercise Identifier	Fixed	Must be as expected.	Error if no match.
PDU Type	Fixed	Constant 3	
Firing Entity ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	Error if no match with Event ID Firing Entity.
Target Entity ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	Informative
Munition ID Site Host Entity	Fixed at time of Fire PDU	Same as corresponding Fire PDU	Terminate Munition Entity
Event ID Site Host Event	Fixed at time of Fire PDU	Same as corresponding Fire PDU	Error if original Event ID not received.
Time Stamp	UTC or internal simulator time of detonation	See Time Stamp for Entity State PDU To accurately determine damage, must know where ITB is at this time. This is absolutely necessary for collateral damage.	Compare to ITB location at time to determine damage.
Location in World X-component Y-component Z-component	Variable, based on location where detonation occurs	Location in world coordinates. See Entity Location for Entity State PDU.	Compare to ITB location at time to determine damage, used in Detonation Assessment task.
Burst Descriptor Munition Warhead Fuze Quantity Rate	Fixed at time of Fire PDU	Same as corresponding Fire PDU. Used in damage determination.	Factors in damage assessment.
Velocity X-component Y-component Z-component	Instantaneous velocity of munition immediately prior to detonation	Linear velocity of munition in world coordinates, 32-bit floating point, conforming to IEEE 754-1985, in units of meters per second.	Factors in damage assessment, used in Detonation Assessment task.
Location in Entity Coordinates X-component Y-component Z-component	Variable, depending on target location	Location of detonation with respect to the target, in the target's entity coordinates. This field will be zero if no Target Entity ID is specified. Important if ITB is the target entity.	Alternative to computing ITB location at time of detonation, used in Detonation Assessment task if available.

Table 2.2.2-4 Receive PDU: Detonation - (Continued)

Detonation Result	Determined at time of PDU	Enumerated type, Appendix F.30.1.3. There must be a function in the ITB to compute damage based on location of detonation, location of ITB, burst descriptor (including munition ID), velocity of munition, and detonation result. The function may cause ITB capability and appearance to change.	Indication of impact or detonation, affects damage assessment, used in Detonation Assessment task.
Number of Articulation Parameters	TBD	Number of Articulation Parameter records to follow.	N/A until ITB has articulated parts.
Articulation Parameters Change indicator ID-attached to Parameter type Parameter value	TBD	See Articulation Parameters for Entity State PDU.	

Table 2.2.2-5 Receive PDU: Collision

Field	Source	Activity/Notes	Use/Destination
Protocol Version	Fixed	Must match ITB capability.	Error if no match
Exercise Identifier	Fixed	Must be as expected.	Error if no match
PDU Type	Fixed	Constant 4	
Issuing Entity ID Site Host Entity	Fixed	See Entity Identifier for Entity State PDU.	Error if issuing entity not in Entity Data Structure.
Colliding Entity ID Site Host Entity	Determined by collision detection algorithm	Entity Identifier of object that the issuer collided with, zero if unknown or terrain. Important if this is ITB cockpit and ITB has also determined that collision occurred. In this case, a damage function must be added to ITB, and it may cause ITB capability and appearance to change.	Ignore unless Entity ID of ITB
Event ID Site Host Event	Fixed at time of PDU	See Event ID of Fire PDU	Informative
Time Stamp	UTC or internal simulation time of collision	See Time Stamp for Entity State PDU	For collision to occur, mutual determination within TBD seconds.
Velocity X-component Y-component Z-component	Instantaneous velocity of issuer at time of collision	Linear velocity of issuer at time of collision in world coordinates, 32-bit floating point, conforming to IEEE 754-1985, in units of meters per second	Factor in damage assessment, used in Collision Assessment task.
Mass	Instantaneous mass of issuer at time of collision	Mass of issuing entity, 32-bit floating point, conforming to IEEE 754-1985, in units of kilograms	Factor in damage assessment, used in Collision Assessment task.
Location with Respect to Entity X-component Y-component Z-component	Variable, depending on target location	Location of collision with respect to the colliding entity in the colliding entity's coordinates. This field will be zero if no colliding Entity ID is specified. This data will help to determine damage.	Factor in damage assessment, used in Collision Assessment task.

Table 2.2.2-6 Receive PDU: Service Request

Field	Source	Activity/Notes	Use/Destination
Protocol Version	Fixed	Must match ITB capability.	Error if no match
Exercise Identifier	Fixed	Must be as expected	Error if no match
PDU Type	Fixed	Constant 5	
Requesting Entity ID Site Host Entity	Fixed	See Entity Identifier for Entity State PDU	Error if entity not in Entity Data Structures. This entity will receive the repair/resupply response
Servicing Entity ID Site Host Entity	Fixed at time of service request	Entity that provides resupply or repair	If ITB specified, must send immediate response
Service Type	Fixed at time of PDU	Enumerated Type, Appendix F.30.1.9	ITB Response: If 1, then Resupply Cancel PDU, if 2 then Repair Complete PDU
Number of Supply Types	Fixed at time of PDU	Number of Supply Quantity fields to follow	N/A
Supply Quantity Entity Kind Domain Country Category Subcategory Specific Extra Quantity	Fixed at time of PDU	Defined in 5.2.17	N/A

2.3 Performance and Interface Requirements

The performance and interface requirements for adding DIS capability to the ITB are given in this section. They are classified as follows:

- External device interface requirements
- DIS/ITB data interchange requirements
- Task performance requirements

2.3.1 External Device Interface Requirements

Access to external devices is required in order to verify DIS-compliance, to connect to other DIS sites and also to use Universal Coordinated Time. The requirements for the interface are given in this section.

2.3.1.1 Interface to Verify DIS Compliance

Automated Forces Testbed software will be used to verify DIS-compliance in the ITB. The software is available at no cost from the Institute for Simulation and Training (IST) and will be executed on an IBM PC-compatible computer. The minimum hardware configuration for executing the Automated Forces Testbed software is as follows.

- IBM-compatible PC, with the following processor:
  - Intel 486 processor
  - Intel 386 processor with floating point co-processor
- 4 Megabytes memory
- 300 Megabyte hard disk
- 3-Com Etherlink-2 card

WL/AAAS-2, the managers of the ITB, will supply all of the hardware except for the disk drive to support DIS-compliance testing. The disk drive will be purchased by Amherst Systems and delivered to the government during Phase II. The Ethernet connector of the PC executing the Automated Forces Testbed software will plug directly into the Ethernet link from the Alliant computer running the ITB software. This is illustrated in Figure 2.3.1.1-1. The Ethernet link will use the UDP/IP (User Datagram Protocol/Internet Protocol) to exchange information.

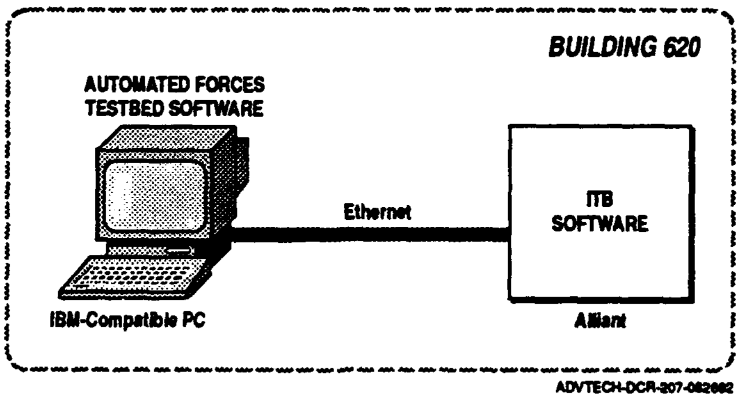

Figure 2.3.1.1-1 Configuration to Verify DIS-Compliance of ITB

2.3.1.2 Connection to Other DIS Sites

For Phase II, Amherst Systems has proposed perform DIS testing with the Flight Dynamics Directorate at Wright-Patterson AFB and also with off-based simulators to be determined at a later date. Figure 2.3.1.2-1 illustrates the on-base network at Wright-Patterson AFB that will support the tests. There are bulk fiber optic cables that run between the Air Force Institute of Technology (AFIT) and Building 620, and also between Building 620 and the Flight Dynamics Directorate. Individual fiber lines will be dedicated to connect the sites. AFIT and the Flight Dynamics Directorate intend to establish a connection within Building 620 that bypasses the ITB, which will give the Flight Dynamics Directorate independent DIS capability. Although the fiber link exists between AFIT and Building 620, the local area networks (LAN's) remain disconnected. To establish connectivity between Building 620 and AFIT, Amherst System has offered to procure and deliver to the government two fiber optics transceivers as part of the Phase II effort. The early procurement of these items will permit debugging of the interface software to be developed under this task. Installation of the fiber optics transceivers also gains access to the T-1 (Defense Simulator Internet) long haul network. T-1 lines are often used for DIS testing. The T-1 access is essential for the testing with off-base simulators as discussed in the Phase II proposal.

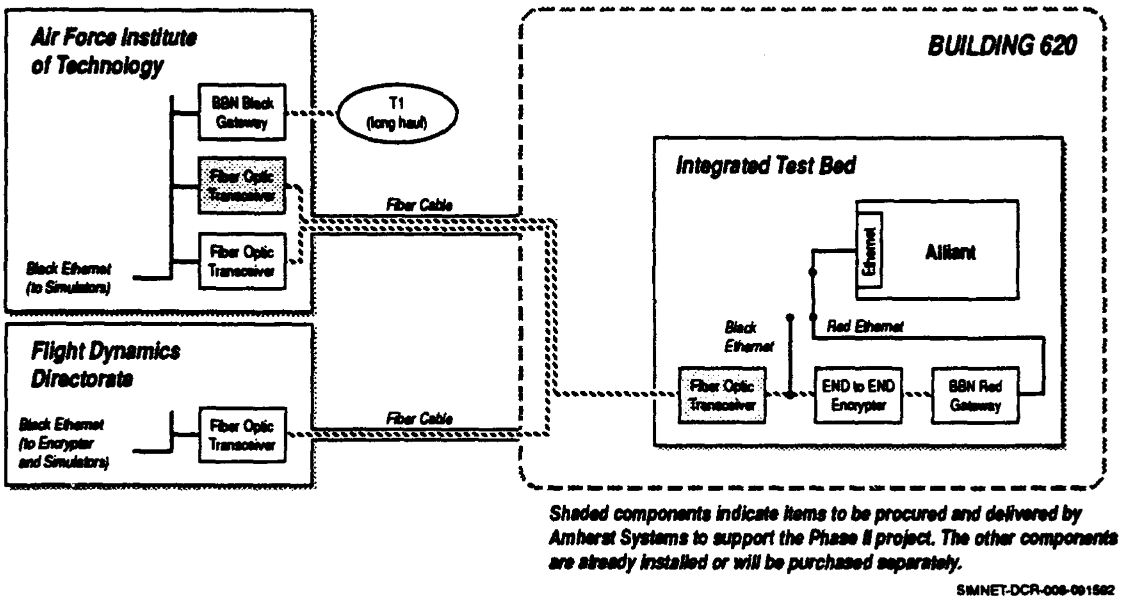

Figure 2.3.1.2-1 Links at Wright-Patterson AFB

2.3.1.3 Use of Universal Coordinated Time (UTC)

Although the ITB presently uses only its computer processor clock, the importance of using Universal Coordinated Time (UTC) for DIS applications is well understood. UTC requires access to GPS signals. Fortunately, UTC capability has been implemented in the Integrated Electromagnetic Simulator System (IESS), which is located in a vault within the ITB facility. The GPS reception antenna and its wiring are already in place in Building 620. Use of UTC will require access to the GPS signal, purchase of a receiver module, integration of the receiver module into the Alliant computer multiprocessor system, and development of software to interpret and apply the time. WL/AAAS-2 is investigating the acquisition and installation of equipment to support UTC. Manufacturers such as Odetics and Datum sell appropriate modules. Should this equipment be purchased and installed, Amherst Systems will be responsible for developing software to exploit the capability during Phase II of this project.

2.3.2 DIS/ITB Data Interchange Requirements

There are two areas of interest in the exchange of information between DIS and the ITB. The first concerns the efficient processing of PDU's and management of the interface to other simulator sites. The second involves the management of information required to support the DIS interface. Each is discussed in the following paragraphs.

2.3.2.1 Management of DIS Interface

The Ethernet interface used to exchange PDU's with other simulators must be managed efficiently in order to minimize late ies and distortions in the simulation. Outgoing PDU's must be transmitted as soon as possible after they are generated. Incoming PDU's must be processed as quickly as possible when they are received. To accomplish efficient processing of PDU's, queues will be established to hold the PDU's.

2.3.2.1.1 Incoming PDU's

Receipt of a PDU will trigger interrupt driven processing in the Ethernet input routine. The input routine will place the PDU at the end of Unprocessed Input PDU Queue. It will then start a task whose purpose is to process the PDU's and distribute information to the appropriate ITB software modules. This task will continue to execute as long as there are unprocessed PDU's queued. Figure 2.3.2.1.1-1 illustrates the processing and Figure 2.3.2.1.1-2 provides pseudo-code for the processing. Even though Figure 2.3.2.1.1-1 shows a large number of PDU's queued, it is expected that the queue will be empty most of the time for small and moderate sized scenarios.

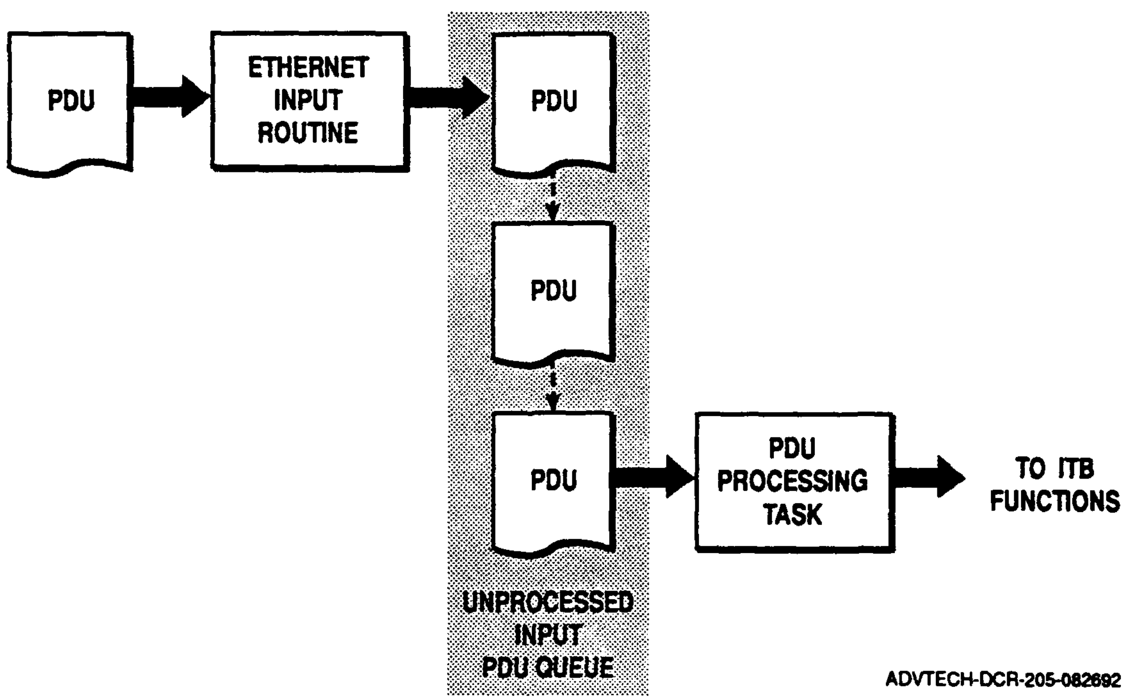

Figure 2.3.2.1.1-1 Management of Incoming PDU's

```
ETHERNET INPUT ROUTINE

Begin
  Add PDU to Queue;
  Start PDU Processing Task;
End;

(Note: This routine is started automatically
   when a PDU is received)
```

```
PDU PROCESSING TASK

Begin
  Loop
    If No PDU Queued Then
      Wait to be Started;
    End If;
    Get Next PDU;
    Process PDU;
  End Loop;
End;
```

ADVTECH-DCR-206-082692

Figure 2.3.2.1.1-2 Pseudo-code for Processing Incoming PDU's

2.3.2.1.2 Outgoing PDU's

PDU's can be generated by a number of independently executing ITB tasks. There are two problems that can result. The first is that PDU's may be generated simultaneously by the tasks, but must be transmitted serially by the interface. The second is that the interface may be busy and not able to accept outgoing messages for a period of time.

To solve the first problem, the outgoing PDU's will be queued in the Unprocessed Output PDU Queue until the output task is ready to accept them. When an ITB software module generates a PDU, the PDU will be appended to a list of all outgoing PDU's awaiting transmission. Computer system interlock operations will be used to prevent conflict in case two or more PDU's are added at the same time.

To ensure that the queued PDU's are transmitted as quickly as possible, an Ethernet Output Task will be used. The will take a PDU from the queue and write it to the Ethernet. The Ethernet Output Task will be started by the tasks that generate PDU's and will continue to execute as long as there are unprocessed PDU's queued.

Figure 2.3.2.1.2-1 illustrates the processing and Figure 2.3.2.1.2-2 provides pseudo-code for the tasks. Even though Figure 2.3.2.1.2-1 shows several PDU's queued, it is expected that the queue will be empty most of the time.

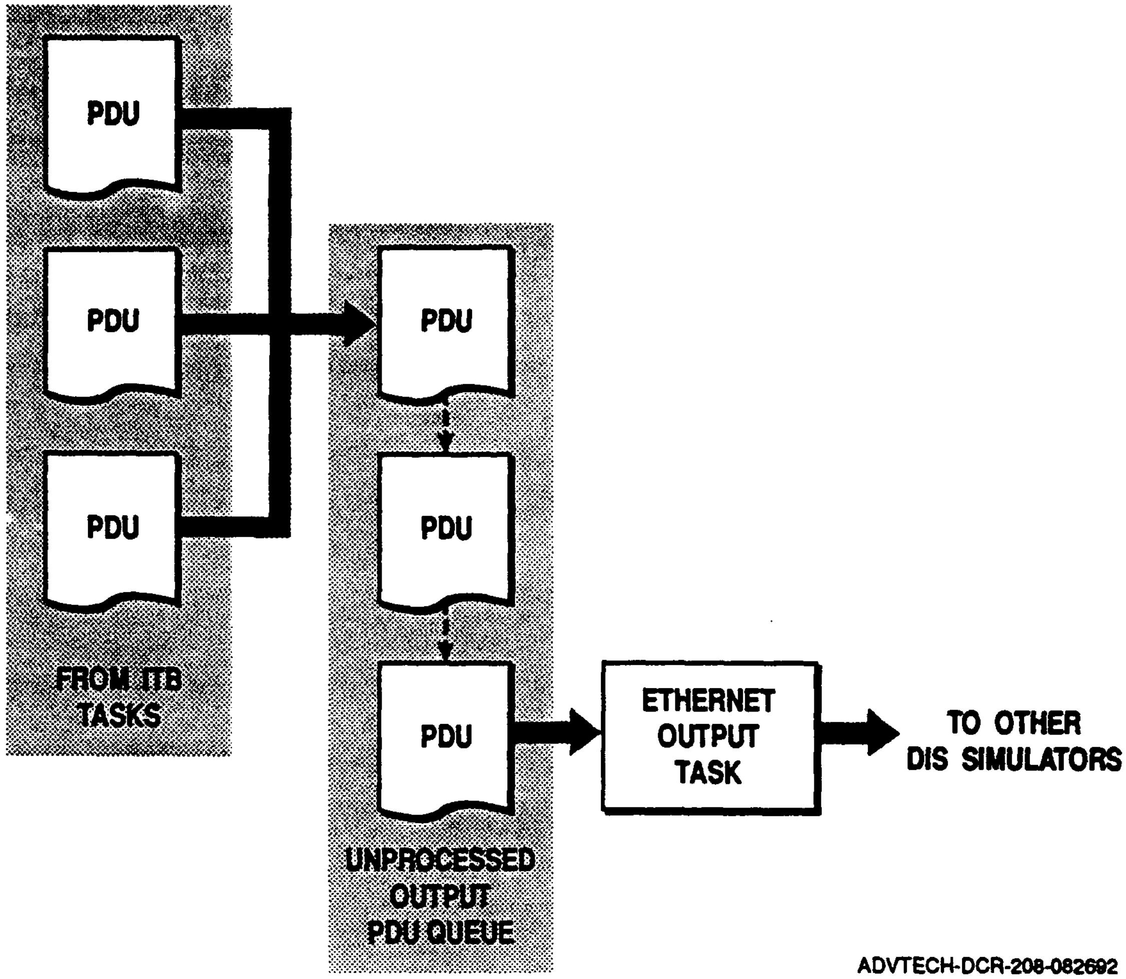

Figure 2.3.2.1.2-1 Management of Outgoing PDU's

```
PDU GENERATION

•
•
•
Generate PDU;
Add PDU to Queue;
Start Ethernet Output Task;
•
•
•
```

```
ETHERNET OUTPUT TASK

Begin
  Loop
    If No PDU Queued Then
      Wait to Be Started;
    End If;
    Get Next PDU;
    Write PDU to Ethernet;
  End Loop;
End;
```

ADVTECH-DCR-209-082692

Figure 2.3.2.1.2-2 Pseudo-code for Processing Outgoing PDU's

2.3.2.2 Management of DIS Data

Information from the incoming DIS PDU's must be assimilated into the ITB processing. Similarly, information about the activity of the ITB simulated aircraft must be put into DIS formats for transmission to the other simulator sites. The requirements for this processing are discussed in this section.

2.3.2.2.1 New ITB Data Structures

To support the new functionality required by DIS, several new data structures will be added to the ITB software. They are as follows:

- Entity Data Structure
- Ordnance Detonation Queue
- Collision Damage Queue

2.3.2.2.1.1 Entity Data Structure

The Entity Data Structure will be a list of Entity Records, indexed by Entity Identifier from the Entity State PDU. Each Record will contain movement and appearance information about the entity. Figure 2.3.2.2.1.1-1 illustrates the layout of an Entity Record.

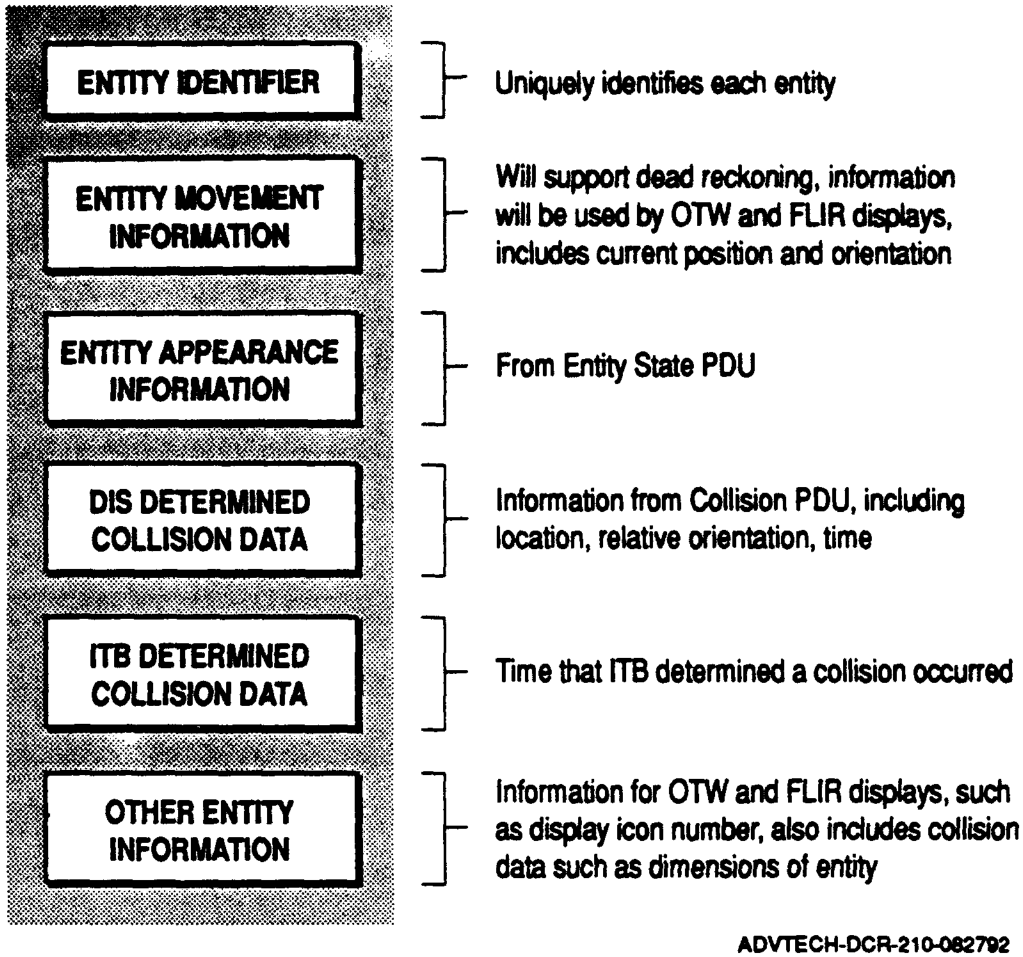

Figure 2.3.2.2.1.1-1 Layout of Entity Record

The Entity Data Structure will be accessed for the following reasons:

a. Receipt of Entity State PDU. When an Entity State PDU is received, a new Entity Record will be created if one does not yet exist. If a new Entity Record is created, other Entity Information will be taken from the Entity Database (see Section 2.4.2.2). Entity Movement Information and Entity Appearance Information will be copied from the Entity State PDU. Dead Reckoning calculations will be performed to bring the entity's position and orientation up to the current time.

b. Dead Reckoning Update. During a dead reckoning update, each Entity Record will be accessed to update the entity's position and orientation to reflect the current time. All of the information required to perform dead reckoning updates will be kept in the Entity Record.

c. OTW/FLIR Update. During an OTW/FLIR update, each Entity Record will be examined. If an entity is within display range, the entity's position and orientation, its appearance and its icon number will be used to construct a message for the OTW/FLIR display.

d. **Receipt of Collision PDU.** When a Collision PDU involving the ITB is received, the time of the ITB Determined Collision Data will be examined. If the time is close, then a collision will be deemed to have occurred and an entry will be placed on the Collision Damage Queue. If there is no time match, then a potential collision has occurred. To retain this information, potential collisions received through DIS PDU's will be retained in the DIS Determined Collision Data. The Entity Record will contain information from the PDU including the timestamp. Periodically, the ITB software will remove data which are obsolete and were not confirmed by the ITB. Data will also be removed if the collision is confirmed by the ITB.

e. **ITB Collision Evaluation.** Periodically, the ITB will check to see if it has collided with another entity. If the ITB calculates that a collision has occurred, the DIS Determined Collision Data will be examined for a collision involving the other entity. If the times are close, then a collision will be deemed to have occurred and an entry will be placed on the Collision Damage Queue. If there is no time match, then a potential collision has occurred. To retain this information, the time of the potential collision computed by the ITB will be retained in the ITB Determined Collision Data. Periodically, the ITB software will remove data which are obsolete and were not confirmed by DIS. Data will also be removed if the collision is confirmed by DIS.

2.3.2.2.1.2 Ordnance Detonation Queue

When a Detonation PDU is received, its information will be placed in the Ordnance Detonation Queue. A Detonation Evaluation task will remove entries from this queue and determine the damage to the ITB, if any, caused by the detonation.

2.3.2.2.1.3 Collision Damage Queue

According to the DIS standard, a collision between entities must be confirmed by both entities involved in the collision. When both the ITB and another DIS entity have determined that a collision between them has occurred, information from the Collision PDU will be placed in the Collision Damage Queue. A Collision Evaluation task will remove entries from this queue and determine the damage to the ITB, if any, caused by the collision.

2.3.2.2.2 Processing of Incoming Information

Information from the incoming PDU's provides the ITB with data about the other players in the simulation. The following types of information are present in the incoming DIS PDU's.

- Entity appearance
- Entity movement
- Weapon firing
- Ordnance detonation
- Collisions
- Repair and resupply requests

Entity appearance and entity movement information will be placed in the Entity Data Structure by the PDU Processing Task. Entity appearance will be reflected in the OTW Display. Entity movement information is used to compute the position and orientation of the other DIS entities. Weapon firing information will be noted but no special activity will result. Ordnance detonation will cause the Detonation Evaluation Task to execute in order to determine effects on the ITB. Collision information received in PDU's will be used to determine damage should the ITB concur that a collision has occurred. Repair and resupply requests will cause the PDU Processing Task to generate PDU responses denying the capability.

2.3.2.2.3 Processing of Outgoing Information

Information for outgoing PDU's provides other DIS entities data about the activity of the ITB. The following types of information are present in the outgoing PDU's.

- ITB appearance
- ITB movement
- Release of ordnance
- Ordnance detonation
- Collisions

ITB appearance may change due to damage from ordnance detonation or collisions and the change will be included in outgoing Entity State PDU's.. ITB movement will be reflected in the Entity State PDU's, and other location/movement information will be included in outgoing Fire, Detonation, and Collision PDU's. In all cases, this information will be converted into ECEF or entity vector formats as appropriate. Release of ordnance by the ITB will cause a Fire PDU to be issued. The ITB routine that computes the ballistic trajectory of the released ordnance will supply detail for the Detonation PDU. Periodically, the code that updates the locations of the DIS entities will check whether a collision with the ITB has occurred. If a collision is detected, the Collision PDU will be generated.

2.3.3 Task Performance Requirements

Table 2.3.3-1 summarizes how often activities related to DIS must be performed by the enhanced ITB software.

Table 2.3.3-1 Task Performance Characteristics

Activity	Frequency	Notes
Process incoming PDU	Immediate, upon receipt	Continuous operation as long as there are unprocessed PDU's
Transmit outgoing PDU	Immediate	Continuous operation as long as there are PDU's waiting to be transmitted
Dead reckoning of DIS entities	16 Hz	Synchronized with OTW/FLIR displays
Check for collisions	1 Hz	Synchronized with dead reckoning
Remove obsolete potential collisions	1 Hz	Synchronized with dead reckoning
Generate Entity State PDU	Immediate	Caused by change in appearance, elapsed time, or movement beyond dead reckoning thresholds
Generate Fire PDU	Immediate	Release of ordnance
Generate Detonation PDU	Immediate	Detonation of ordnance
Generate Collision PDU	Immediate	When ITB software determines a collision has occurred
Generate Repair/Resupply PDU's	Immediate	Upon receipt of Service Request PDU
Compute detonation damage	Immediate	When entry added to Ordnance Detonation Queue, synchronized to aircraft modeling
Compute collision damage	Immediate	When entry added to Collision Damage Queue, synchronized to aircraft modeling

2.4 Top Level Design

Figure 2.4-1 illustrates operation of the ITB using DIS to exchange information. The activity of the other players is received through the DIS interface instead of being generated internally by the ITB. The activity of the ITB is transmitted to the other players also using the DIS interface. The use of DIS should be transparent to the ITB "pilot."

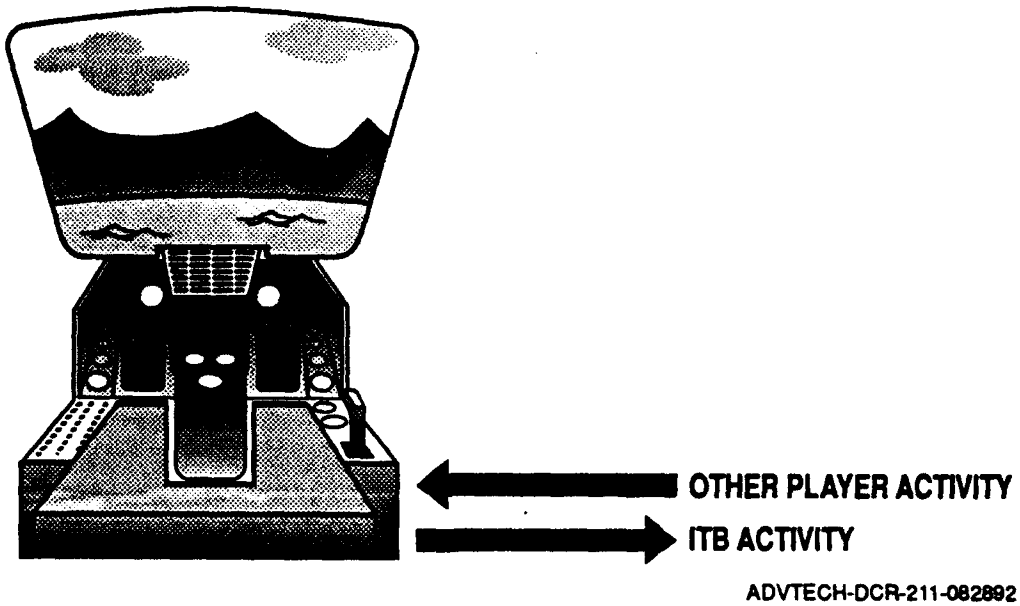

Figure 2.4-1 ITB Activity Using DIS

The top level design for the implementation of DIS capability in the ITB reflects the intuitive notion of transparency of DIS operation. Figure 2.4-2 illustrates the tasks and the major data flows. The "pilot" interacts with the ITB through the use of the cockpit controls and displays. The OTW/FLIR and HUD displays provide additional information about the environment. None of this will change when DIS capability is implemented. Instead, the new DIS information will operate on the internal data used to support the ITB internal functions and displays.

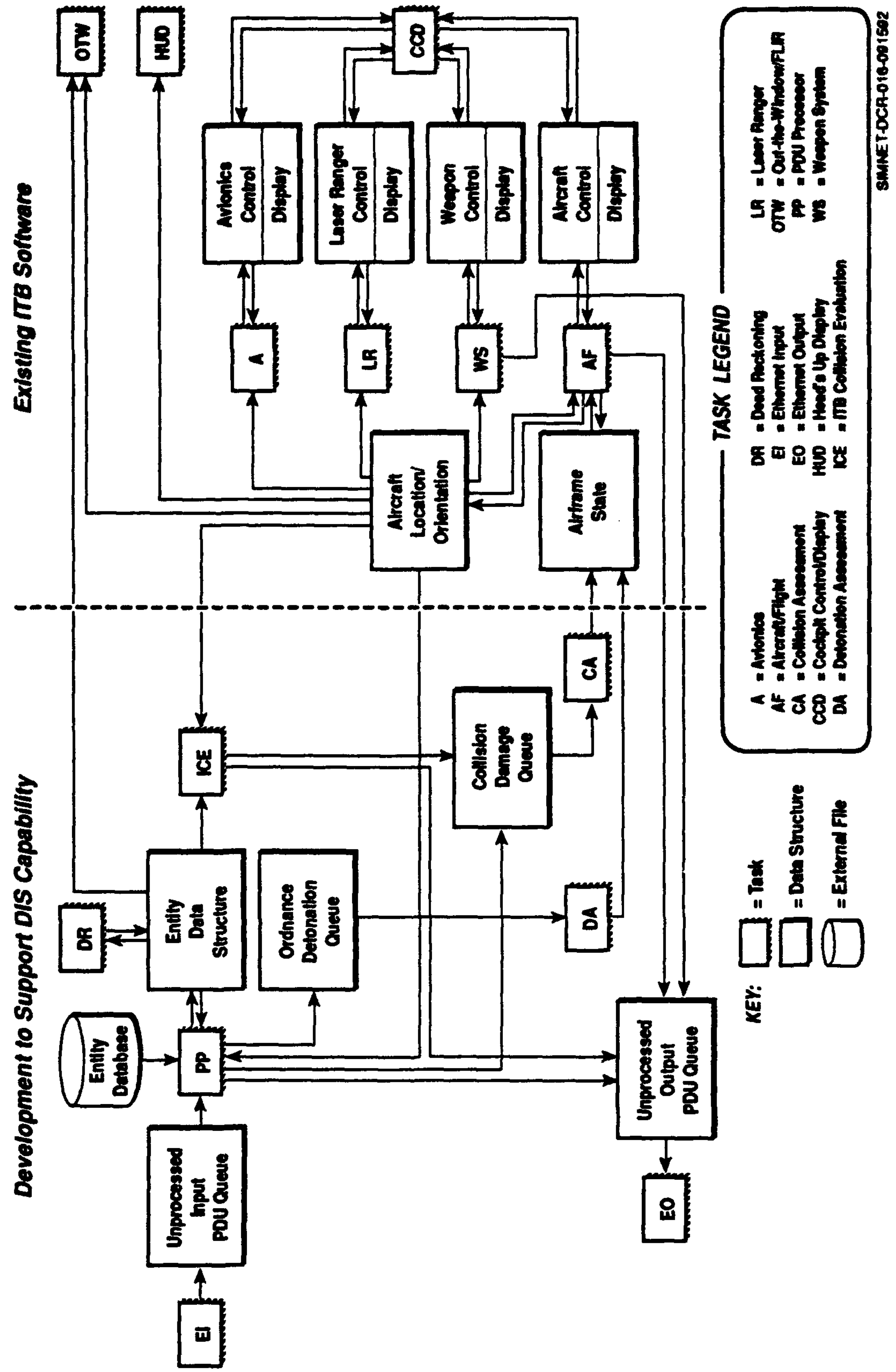

Figure 2.4-2 ITB/DIS Task Interrelationships

2.4.1 Task Functions

The ITB software consists of a number of independently executing tasks. Figure 2.4-2 illustrates the functional relationship of these tasks. In order to support DIS capability, some of the existing ITB tasks must be modified, and other new tasks must be developed. The new/modified functionality of these tasks is outlined in this section.

2.4.1.1 Modifications to Existing ITB Tasks

This section describes the modifications that must be made to the existing ITB tasks in order to support DIS capability.

2.4.1.1.1 Aircraft/Flight (AIRPLN, EARTH)

The task must generate an Entity State PDU whenever the thresholds in location, orientation or time have been exceeded, or a change in appearance has occurred. Currently in the ITB code, task EARTH updates the aircraft location and task AIRPLN updates the orientation, velocities and accelerations. The EARTH task will be integrated into AIRPLN to simplify these operations. The AIRPLN task executes 64 times per second. Each time it executes, it will perform a dead reckoning update of itself. The actual location and orientation will be compared to the dead reckoned location and orientation. Should the difference exceed thresholds established during initialization, an Entity State PDU will be generated. An Entity State PDU will be generated should the length of time since the last Entity State PDU exceed a threshold established during initialization. An Entity State PDU will also be generated when the aircraft appearance changes. This can be caused by the new Collision Assessment and Detonation Assessment tasks to support the DIS capability. These tasks will calculate damage to the aircraft due to collision or detonation of munitions. When a PDU is generated, it will be placed in the Unprocessed Output PDU Queue.

Reference to terrain information may need to change. ITB software is hard-coded for a specific area. Changes must be made to support arbitrarily specified gaming areas.

2.4.1.1.2 Avionics (Several tasks)

Avionics tasks will be virtually unchanged. References to terrain information for devices such as the altimeter may need to change to support arbitrarily specified gaming areas.

2.4.1.1.3 Cockpit Control/Display (Several tasks)

Cockpit control and display tasks will not be changed by this effort.

2.4.1.1.4 Head's-Up Display (HUDBBN)

The Head's-Up Display will remain virtually unchanged. The only modification may be to alter the aircraft location and orientation values slightly to compensate for differing earth models and terrain interpretations. The earth and terrain models underlying the HUD are not known as of this writing.

2.4.1.1.5 Laser Ranger (LSR)

The Laser Ranger task will be virtually unchanged. References to terrain information may change to support arbitrarily specified gaming areas.

2.4.1.1.6 Out-the-Window/FLIR Display (BBN_INTERFACE)

The OTW/FLIR Display task must be modified to display the other DIS entities. The Entity Data Structure will contain all of the information necessary to produce the display. The task will convert each entity's location, orientation, display icon, and size into a message that will be included in the packet to be sent to the display. The location and orientation must be converted from the ECEF coordinates in the Entity Data Structure to the latitude/longitude/altitude required by the display.

The ITB aircraft location and orientation values may be altered slightly to compensate for differing earth models and terrain interpretations.

2.4.1.1.7 Weapon System (WEAPEX)

The Weapon System task must be modified to generate a Fire PDU at the moment when an MK-82 bomb is released. This occurs at the location in program WEAPEX where subroutine SCORE is called. Location and velocity information for the PDU will be based on the aircraft information. The Fire PDU will be placed in the Unprocessed Output PDU Queue.

The Weapon System task must also be modified to generate a Detonation PDU at the appropriate time. Presently, subroutine SCORE is called to determine miss distance from the intended target. Subroutine TRAJEC computes the critical information of location and time of detonation. Since the flight path of the bomb is ballistic after it is released, this information can be computed at the time that the Fire PDU is generated. When DIS is used, routine TRAJEC will be called directly. Information from the routine will be saved until the computed time of detonation, at which time the Detonation PDU will be generated. The Detonation PDU will be placed in the Unprocessed Output PDU Queue.

2.4.1.2 New ITB Tasks

This section describes the new tasks that must be added to ITB software in order to support DIS capability. Seven new tasks are required. Each of the seven tasks will be started by the ADSS Executive and perform initialization functions before simulation begins. None of the tasks execute continuously. Some execute cyclically at a fixed rate, triggered by the ADSS Executive. Others are data driven, executing only when there is data available. Table 2.4.1.2-1 summarizes these task attributes.

Table 2.4.1.2-1 Execution Attributes of New Tasks

New Task	Operation
Collision Assessment	Data driven, when data in the Collision Damage Queue.
Dead Reckoning	16 Hz, controlled by the executive.
Detonation Assessment	Data driven, when data is in the Ordnance Detonation Queue.
Ethernet Input	Data driven, when PDU received over Ethernet.
Ethernet Output	Data driven, when data is in the Unprocessed Output PDU Queue.
ITB Collision Evaluation	1 Hz, controlled by the executive.
PDU Processing	Data driven, when data is in the Unprocessed Input PDU Queue.

2.4.1.2.1 Collision Assessment

The Collision Assessment task will determine damage caused to the ITB after collision with another entity. This task executes only when a confirmed collision has occurred, that is, when both the ITB and the other entity have independently determined that a collision has occurred. The damage, if any, will manifest itself in changes to the airframe and aircraft appearance. The resultant Entity State PDU will be generated by the Aircraft Control task.

The Collision Assessment task will execute only when there are entries in the Collision Damage Queue. Both the ITB Collision Evaluation task and the PDU Processor can place items in the queue. The Collision Assessment task will be started if it is not already executing when this occurs. It will remove the queued damage information, which consists of pertinent information from the Collision PDU's. The information used to determine damage includes the mass of the entities, their velocities, and the points of contact of each entity. The actual damage evaluation algorithm to be used has not yet been determined, and will be deferred until Phase II.

2.4.1.2.2 Dead Reckoning

The Dead Reckoning task will be responsible for performing dead reckoning updates for all of the external DIS entities. Dead reckoning information is included in each Entity State PDU and will be stored in the Entity Data Structure. This task will execute at a 16 Hz rate and will be synchronized with the Out-the-Window/FLIR Display task so that these displays always have current information. Dead reckoning calculations will be performed with ECEF/Entity coordinates using the formulas from the document "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation," Appendix I. Other tasks that use the dead reckoned information will be responsible for converting coordinates into those appropriate for their application.

2.4.1.2.3 Detonation Assessment

The Detonation Assessment task will determine damage caused to the ITB after detonation of ordnance. The task executes only when a Detonation PDU has been received. The damage, if any, will manifest itself in changes to the airframe and aircraft appearance. The resultant Entity State PDU will be generated by the Aircraft Control task.

The Detonation Assessment task will execute only when there are entries in the Ordnance Detonation Queue. The Detonation Assessment task will be started if it is not already executing when this occurs. It will remove the queued information, which consists of pertinent information from the Detonation PDU. The information used to determine damage includes the type of ordnance, the relative location of the detonation to the ITB, and the orientation of the ITB. The actual damage evaluation algorithm to be used has not yet been determined, and will be deferred until Phase II.

2.4.1.2.4 Ethernet Input

The Ethernet Input task will be responsible for efficient management of the Ethernet local area network (LAN). Since DIS typically operates using an unreliable data exchange protocol, it is important that the input task be capable of accepting input at any time. To minimize ITB processing delays, the task must place PDU's in the Unprocessed Input PDU Queue as quickly as possible. The Ethernet Input task will be coded with both of these concerns taken into consideration. The task will be started by the Alliant computer's operating system whenever a PDU is received. It will continue to execute as along as PDU's continue to be received. "Read ahead" buffers will be used so that PDU's will not be dropped should the task be busy when a PDU arrives. The Ethernet Input task starts the PDU Processor task when a PDU is placed in the Unprocessed Input PDU Queue.

It is assumed that the Alliant computer system is capable of implementing whatever OSI standard is used for communication. Amherst Systems will code any software above the driver level that is required to implement the desired protocol. If it turns out that the Ethernet driver software or hardware is not capable of supporting I/O, additional procurement by WL/AAAS-2 may be required.

2.4.1.2.5 Ethernet Output

The Ethernet Output task will output ITB generated PDU's as quickly as possible over the Ethernet LAN. PDU's can be generated by a number of ITB tasks. Simultaneous generation of PDU's and the possibility that the Ethernet could be busy means that it may not be possible to transmit a PDU immediately after it is created. Instead of making time critical tasks wait until the PDU is transmitted, the PDU's will be placed in a queue. The Ethernet Output task will remove PDU's from the queue and transmit them over the Ethernet to other DIS sites. The Ethernet Output task will be started whenever a PDU is added to the queue. It will continue to execute as long as PDU's remain in the queue.

It is assumed that the Alliant computer system is capable of implementing whatever OSI standard is used for communication. Amherst Systems will code any software above the driver level that is required to implement the desired protocol. If it turns out that the Ethernet driver software or hardware is not capable of supporting I/O, additional procurement by WL/AAAS-2 may be required.

2.4.1.2.6 ITB Collision Evaluation

The ITB Collision Evaluation task will determine whether the ITB has collided with another entity. It will be executed at a rate to be determined. The task will compare the ITB location, orientation, shape, and size with all other entities' data using an algorithm to be developed in Phase II. If the ITB determines that a collision has occurred, it will generate a Collision PDU, add it to the Unprocessed Output PDU Queue, and start the Ethernet Output task if it is not yet executing. Additionally, the entity record for the colliding entity will be examined to see if a concurrent PDU has been received confirming the collision. If so, information from the record will be placed in the Collision Damage Queue, the Collision Assessment task will be started if it is not yet executing, and collision information in the record will be cleared. If there is no confirming collision data, the time of the evaluation will update the record.

2.4.1.2.7 PDU Processor

The PDU Processor task processes each of the PDU's received from other DIS sites. The PDU Processor is started by the Ethernet Input task. It takes a PDU from the Unprocessed Input PDU Queue and performs processing as described below. The PDU Processor task will continue to execute as long as PDU's remain in the Unprocessed Input PDU Queue.

a. Entity State PDU: The Entity Data Structure will be examined to see if the entity already exists. If the entity does not yet exist, the following special processing will be performed.

    - A new record for the entity will be created in the Entity Data Structure.
    - Information about the entity type will be retrieved from the Entity Database and added to the record.

    The record in the Entity Data Structure will receive updated movement information from the PDU. The location and orientation placed into the record will reflect a dead reckoned update from the time of the PDU to the time of the processing.

b. Fire PDU: The entity record in the Entity State PDU will be updated to reflect the fact that the entity has fired.

c. Detonation PDU: Information about the munition, time of detonation and location of detonation will be placed in the Ordnance Detonation Queue. If the munition is represented as a separate entity, a flag will be set in its record in the Entity Data Structure to indicate that it has detonated. The Detonation Assessment task will be started if it is not currently executing.

d. Collision PDU: The entity record for the colliding entity will be examined to see if the ITB has concurrently determined that a collision has occurred. If so, all collision information in the record will be cleared, information from the Collision PDU will be placed in the Collision Damage Queue, and the Collision Assessment task will be started if it is not yet executing. If there is no concurrent collision, information from the Collision PDU will update the record.

e. Service Request PDU: If the service request is for supplies, a Resupply Cancel PDU will be generated. If the service request is for repairs, a Repair Complete PDU (with "no repairs performed") will be generated. The PDU will be placed in the Unprocessed Output PDU Queue and the Ethernet Output task will be started if it is not currently executing.

f. Other PDU's: All other PDU's will be ignored.

2.4.2 Special Areas of Investigation

This section discusses areas of significant technical importance for the successful completion of Phase II work.

The following paragraphs will describe the relationships between the coordinate systems used by both DIS and the ITB aircraft model. Each paragraph will be divided into three sections. The first section will describe the position and movement data from the perspective of the ITB cockpit (i.e. what does the ITB have that is required by the DIS network?). The second section will describe the position and movement data from the perspective of the DIS network (i.e. what does the DIS network have that is required by the ITB cockpit?). The third section will provide the solution (or methodology) of transferring the position and movement data between the ITB and DIS worlds. Other coordinate transforms, such as those that apply to both the Out-The-Window (OTW) and Heads-Up-Display (HUD) displays can be performed in a similar manner. Currently, however, we do not have all of the data necessary to completely define these transformations. Properly solving these transformations will be an important part of the Phase II effort.

The following provides a list of the Entity State PDU position and movement data items of interest. **Readers Note:** The information contained within the parenthesis references paragraphs from the document entitled "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation" Military Standard Final Draft 30 October 1991.

1. Coordinate System Geometry - Inertial/World Coordinate Reference (5.2.21)

2. Coordinate System Geometry - Entity Coordinate Reference (5.2.20.1)

3. Linear Velocity Vector (5.2.20.3)

4. Linear Acceleration Vector (5.2.20.2)

5. Angular Velocity Vector (5.2.1 and 5.2.2)

6. Entity Location (5.2.21)

7. Entity Orientation/Euler Angles (3.14, 5.2.1 and 5.2.11)

8. Dead Reckoning (5.2.2, 5.2.20.2 and Appendix I)

2.4.2.1.1 Coordinate System Geometry - Inertial/World Coordinate Reference

When formulating and solving problems in flight dynamics, a number of frames of reference (coordinate axis) must be used for specifying various parameters. Techniques must also exist to allow for the transformation of quantities from one coordinate system to another. In order for Newtons second law of motion (F=ma) to hold true at all times, one of these systems must be fixed to the earth.* Such a system is denoted as the Inertial (or World) Coordinate Reference System.

The next few paragraphs describe the ITB North-East-Vertical (NEV) Inertial Coordinate reference system, the DIS Earth-Centered-Earth-Fixed (ECEF) Inertial Coordinate reference system and the methodology required to translate between the two systems.

2.4.2.1.1.1 ITB North-East-Vertical (NEV) Reference System

In many problems of airplane dynamics, the rotation of the Earth can be neglected and, as such, any reference frame fixed to the Earth can be used as an inertial frame. The ITB simulator employs such an inertial coordinate system (named North-East-Vertical (NEV)). NEV is an Earth-surface frame with its origin fixed at the end of the runway where the ITB simulator is initialized. The following applies to the ITB NEV coordinate system:

1. **X Axis:** Positive direction points to the North Pole.

2. **Y Axis:** Positive direction points to the East.

3. **Z Axis:** Positive direction points along the local gravity vector and perpendicular to the surface of the Earth. Since the ITB Earth model is an ellipsoid (see the definition of the ITB Earth model and Figure 2.4.2.1.1.3-2), this vector (for the most part) does not go through the center of the Earth. The exceptions to this rule occur at the Equator and the Poles.

4. **Origin:** Fixed at the end of the runway where the ITB simulator is initialized.

5. **Earth Model:** World Geodetic System 1984 (WGS-84) ellipsoid with a semi-major axis (a) of 6,378,137 meters and a semi-minor axis (b) of 6,356,752.3142 meters.

An illustration of the NEV coordinate system is provided in Figure 2.4.2.1.1.3-1.

---

* If a is the acceleration relative to a reference frame that has either a rotation and/or acceleration of its origin, Newton's second law (F=ma) does not hold, as additional terms that depend upon the motion of the reference frame have to be added to the equation.

2.4.2.1.1.2 DIS Earth-Centered-Earth-Fixed (ECEF) Reference System

In order to account for the rotation of the Earth, the DIS standard requires that the origin of its inertial frame be fixed at the centroid of the Earth (ECEF). The following applies to the DIS ECEF inertial coordinate system:

1. **X Axis:** Positive direction passes through the Prime Meridian at the Equator (0 Degrees Latitude, 0 Degrees Longitude).
2. **Y Axis:** Positive direction passes through 90 degrees East Longitude at the Equator (0 Degrees Latitude, 90 Degrees East Longitude).
3. **Z Axis:** Positive direction passes through the North Pole.
4. **Origin:** Fixed at the centroid of the Earth.
5. **Earth Model:** World Geodetic System 1984 (WGS-84) ellipsoid with a semi-major axis (a) of 6,378,137 meters and a semi-minor axis (b) of 6,356,752.3142 meters.

An illustration of the ECEF coordinate system is provided in Figure 2.4.2.1.1.3-1.

2.4.2.1.1.3 NEV-ECEF Coordinate Transformation Methodology

As was discussed in the preceding paragraphs, there are inherent differences between the ITB NEV inertial coordinate system (whereby the origin is fixed on the surface of the Earth) and the DIS ECEF inertial coordinate system (whereby the origin is fixed at the center of the Earth). In order to resolve these differences and properly represent the ITB cockpit in the DIS world and, in turn, DIS Entities in the ITB world, an NEV-ECEF transformation matrix has to be developed. Figure 2.4.2.1.1.3-1 illustrates the relationship between the NEV and ECEF coordinate systems.

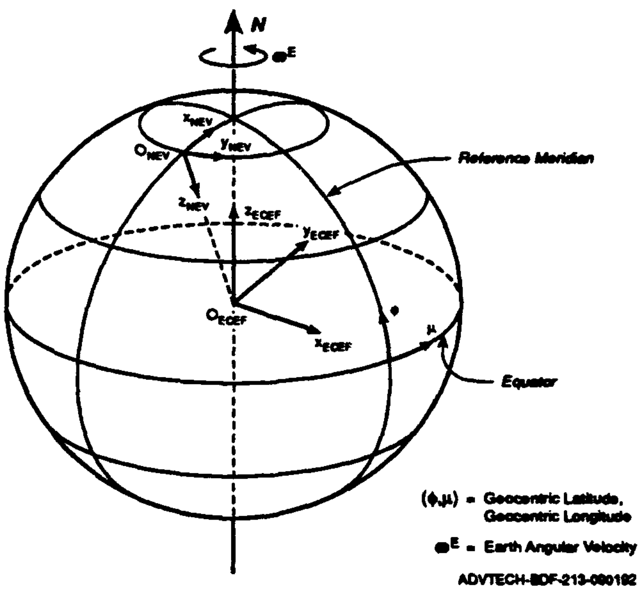

Figure 2.4.2.1.1.3-1 NEV - ECEF Coordinate Systems

The transformations involve the following variables:

1. Geocentric Latitude (ϕ): That latitude, positive as measured from the Equator to the North Pole, assuming a spherical earth.

2. Geodetic Latitude (λ) That latitude, positive as measured from the Equator to the North Pole, assuming a WGS-84 elliptical Earth. This is the latitude as supplied by the ITB simulator.

3. Earth Radius (r_ϕ): The radius of the Earth as a function of the Geocentric Latitude

4. Geocentric/Geodetic Longitude (μ) That longitude positive as measured from (0 Latitude, 0 Longitude) to (0 Latitude, 90 East Longitude). Note that the Geocentric Longitude is equal at all times to the Geodetic Longitude.

A brief description of the NEV to ECEF coordinate transfer methodology is as follows:

1. Determine the relationship between a Geocentric (spherical) and Geodetic (elliptical) Earth in terms of their respective Latitude angles ($\phi=f(\lambda)$ and $\lambda=f(\phi)$).

2. Determine the radius of the Geodetic Earth as a function of its Geocentric Latitude ($r_\phi=f(\phi)$).

3. Determine the homogeneous translation/rotation matrix to convert from an NEV inertial coordinate system to an ECEF inertial coordinate system:

   a. Provide the standard Euler Angle rotation matrices.

   b. Rotate the NEV coordinate system about the Y_{ECEF} axis in order to align the NEV and ECEF origins.

   c. Rotate the NEV coordinate system about the Y_{ECEF} axis in order to compensate for the Geodetic altitude vector.

   d. Translate the NEV coordinate system along the X_{ECEF} axis to the previously calculated radius of Geodetic Earth (r_ϕ).

   e. Rotate the NEV coordinate system about the Y_{ECEF} axis to the Geocentric Latitude.

   f. Rotate the NEV coordinate system about the Z_{ECEF} axis to the Geocentric/ Geodetic Longitude.

   g. Concatenate all of the rotations and translations to form the homogeneous NEV-ECEF transformation matrix.

Figure 2.4.2.1.1.3-2 illustrates the relationship between the Geocentric and Geodetic Earth. Using this figure as well as Figure 2.4.2.1.1.3-1, the next few paragraphs describe the transfer methodology in detail.

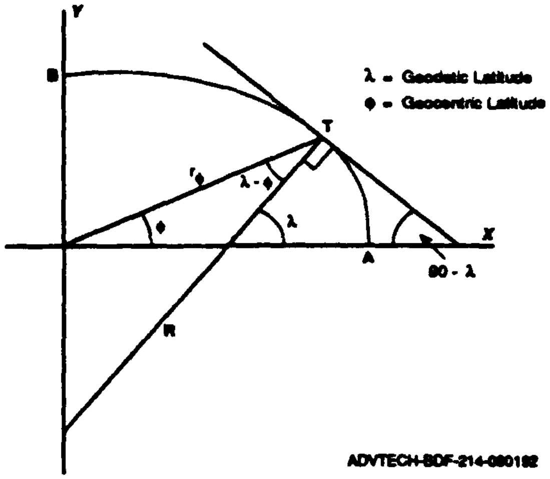

Figure 2.4.2.1.1.3-2 Geodetic to Geocentric Conversion

Step 1: Using Figure 2.4.2.1.1.3-2, the relationship between the Geocentric and Geodetic Earth in terms of their respective latitude angles ($\phi=f(\lambda)$ and $\lambda=f(\phi)$) can be derived as follows:

The equation of an ellipse: $$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \qquad (1)$$

Taking partial derivatives: $$\frac{2x\partial x}{a^2}+\frac{2y\partial y}{b^2}=0 \qquad (2)$$

Rearranging: $$\frac{\partial y}{\partial x}=-\frac{b^2}{a^2}\left(\frac{x}{y}\right) \qquad (3)$$

The slope of the tangent at T: $$\begin{aligned} &= \partial y / \partial x \\ &= -\tan(90-\lambda) \\ &= -\operatorname{cotan}\lambda \\ &= -1/\tan\lambda \end{aligned}$$

or combining: $$\frac{\partial y}{\partial x}=-\frac{1}{\tan\lambda} \qquad (4)$$

The slope of R: $= \tan\lambda$

From (4): $= -\partial x / \partial y$

From (3): $= +\frac{a^2}{b^2}\left(\frac{y}{x}\right)$

Combining: $$\tan\lambda = +\frac{a^2}{b^2}\left(\frac{y}{x}\right) \qquad (5)$$

The slope of r: $$\tan\phi = \frac{y}{x} \qquad (6)$$

Therefore from (5) and (6): $$\tan\lambda = \frac{a^2}{b^2}\tan\phi \qquad (7)$$

From (7): $$\tan\phi = \frac{b^2}{a^2}\tan\lambda \qquad (8)$$

From (7): $$\underline{\underline{\lambda = \arctan\left(\frac{a^2}{b^2}\tan\phi\right)}} \qquad (9)$$

From (8): $$\underline{\underline{\phi = \arctan\left(\frac{b^2}{a^2}\tan\lambda\right)}} \qquad (10)$$

Step 2: Using Figure 2.4.2.1.1.3-2, the radius of the Geodetic Earth as a function of its Geocentric Latitude ($r_\phi = f(\phi)$) can be derived as follows:

The equation of an ellipse:	$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$	(1)
Rearranging:	$a^2b^2 = b^2x^2 + a^2y^2$	(2)
Pythagorean relation:	$r_\phi^2 = x^2 + y^2$	(3)
Also:	$y = r_\phi \sin\phi$	(4)
Squaring (4):	$y^2 = r_\phi^2 \sin^2\phi$	(5)
Substituting (4) into (3):	$r_\phi^2 = x^2 + r_\phi^2 \sin^2\phi$	(6)
Rearranging (6):	$x^2 = r_\phi^2 - r_\phi^2 \sin^2\phi$	(7)
Substituting (7) & (5) into (2):	$a^2b^2 = b^2\left(r_\phi^2 - r_\phi^2\sin^2\phi\right) + a^2r_\phi^2\sin^2\phi$	(8)
Solving (8) for r_ϕ:	$r_\phi = \dfrac{a}{\sqrt{1 + \left(\frac{a^2}{b^2} - 1\right)\sin^2\phi}}$	(9)

Step 3: Using Figure 2.4.2.1.1.3-1 determine the translation/rotation matrix to convert from an NEV inertial coordinate system to an ECEF inertial coordinate system. For the most part in this step, homogeneous matrices will be used. Homogeneous matrices are four-by-four matrices that contain both the rotation and translation components. The first 3 rows by 3 columns (9 elements) contain the rotation component; the last column, first 3 rows (3 elements) contain the translation component. Note that for the rotation component, the standard Euler Angle rotation matrices will be used.

a. **Euler Angle Rotation Matrices:** The orientation of any reference frame relative to another can be given by three angles which are the consecutive rotations about the Z axis through an angle ψ, about the Y axis through an angle θ, and about the X axis through an angle ϕ (in that order). These angles are used to carry one frame of reference in coincidence with another. In flight dynamics, the Euler Angles are those which rotate the body ([illegible]) axis system into coincidence with the inertial reference frame. The Yaw, Pitch, and Roll matrices can be given as follows:

$$\text{Rotation About the Z Axis (Yaw)} = \begin{bmatrix} \text{Cos}\psi & -\text{Sin}\psi & 0 \\ \text{Sin}\psi & \text{Cos}\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Rotation About the Y Axis (Pitch)} = \begin{bmatrix} \text{Cos}\theta & 0 & \text{Sin}\theta \\ 0 & 1 & 0 \\ -\text{Sin}\theta & 0 & \text{Cos}\theta \end{bmatrix}$$

$$\text{Rotation About the X Axis (Roll)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \text{Cos}\phi & -\text{Sin}\phi \\ 0 & \text{Sin}\phi & \text{Cos}\phi \end{bmatrix}$$

b. **Rotate NEV about Y_{ECEF} to Align NEV-ECEF Origins:** Using the matrices developed in (a) above, we must first convert the NEV origin coordinates to ECEF origin coordinates. In order to align the coordinate systems, one rotation of -90 degrees about the Y_{ECEF} axis is required. The matrix M_{EO}^{NO} describes the origin rotation of the NEV coordinate system about the Y_{ECEF} axis and relative to the fixed origin of the ECEF coordinate system:

$$M_{EO}^{NO} = M_R(Y_{ECEF}, -90)$$

$$M_{EO}^{NO} = \begin{bmatrix} \text{Cos}(-90) & 0 & \text{Sin}(-90) & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sin}(-90) & 0 & \text{Cos}(-90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

c. **Rotate NEV about Y_{ECEF} to Compensate for the Geodetic Altitude Vector:** Using the matrices developed in (a) above, we must now compensate for the misalignment of the NEV normal and the vector through the Earth's center. This difference appears as a result of the inherent differences between the spherical and elliptical Earth models (refer back to Figure 2.4.2.1.1.3-2). In essence, the NEV coordinate system must be rotated about the Y_{ECEF} axis through the difference between the Geodetic and geocentric Latitude angles (ϕ-λ) such that the final result produces a Z_{NEV} axis aligned with R. The matrix $M_{E\lambda}^{N\lambda}$ describes the rotation of the NEV coordinate system about the Y_{ECEF} axis and relative to the fixed origin of the ECEF coordinate system:

$$M_{E\lambda}^{N\lambda} = M_R(Y_{ECEF}, \phi - \lambda)$$

$$M_{E\lambda}^{N\lambda} = \begin{bmatrix} \text{Cos}(\phi - \lambda) & 0 & \text{Sin}(\phi - \lambda) & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sin}(\phi - \lambda) & 0 & \text{Cos}(\phi - \lambda) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

d. **Translate NEV Along X_{ECEF} to Radius of Geodetic Earth (r_ϕ):** We must now translate the entire NEV coordinate system along the X_{ECEF} axis by the radius of the Geodetic Earth (r_ϕ) as derived in Step 2 above. The matrix M_{EX}^{NX} describes the translation of the NEV coordinate system along the X_{ECEF} axis relative to the fixed origin of the ECEF coordinate system:

$$M_{EX}^{NX} = M_T(Z_{ECEF}, r_\phi(\phi))$$

$$M_{EX}^{NX} = \begin{bmatrix} 1 & 0 & 0 & r_\phi \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

e. **Rotate NEV about Y_{ECEF} to Geocentric Latitude:** Using the matrices developed in (a) above, we must now orient the Z_{NEV} axis perpendicular to the surface of the Earth. In essence, the NEV coordinate system must be rotated about the Y_{ECEF} axis through the Geocentric Latitude angle ($-\phi$) such that the final result produces a Z_{NEV} axis perpendicular to the surface of the Earth at the Geocentric Latitude of the point in question. The matrix $M_{E\phi}^{N\phi}$ describes the rotation of the NEV coordinate system about the Y_{ECEF} axis relative to the fixed origin of the ECEF coordinate system:

$$M_{E\lambda}^{N\lambda} = M_R(Y_{ECEF}, -\lambda)$$

$$M_{E\lambda}^{N\lambda} = \begin{bmatrix} \text{Cos}(-\lambda) & 0 & \text{Sin}(-\lambda) & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sin}(-\lambda) & 0 & \text{Cos}(-\lambda) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

f. **Rotate NEV about Z_{ECEF} to Geocentric/Geodetic Longitude:** Using the matrices developed in (a) above, we must now orient the X_{NEV} axis North and the Y_{NEV} axis East. In essence, the NEV coordinate system must be rotated about the Z_{ECEF} axis through the Geocentric/Geodetic Longitude angle (μ) such that the final result produces a X_{NEV} axis that points North, a Y_{NEV} axis that points East and a Z_{NEV} axis that points down (and perpendicular to the surface of the Earth) at the Geocentric/Geodetic Longitude of the point in question. The matrix $M_{E\mu}^{N\mu}$ describes the rotation of the NEV coordinate system about the Z_{ECEF} axis relative to the fixed origin of the ECEF coordinate system:

$$M_{E\mu}^{N\mu} = M_R(Z_{BCEF}, \mu)$$

$$M_{E\mu}^{N\mu} = \begin{bmatrix} \text{Cos}(\mu) & -\text{Sin}(\mu) & 0 & 0 \\ \text{Sin}(\mu) & \text{Cos}(\mu) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

g. **NEV Relative to ECEF Matrix:** To finalize the process, we must now concatenate all of the matrices developed in (b) through (f). Therefore, the NEV Relative to ECEF Matrix can be given as:

$$M_{BCEF}^{NEV} = M_{E\mu}^{N\mu} \text{ x } M_{E\phi}^{N\phi} \text{ x } M_{EX}^{NX} \text{ x } M_{E\lambda}^{N\lambda} \text{ x } M_{EO}^{NO}$$

Or, explicitly;

$$M_{BCEF}^{NEV} = \begin{bmatrix} \text{Cos}(\mu) & -\text{Sin}(\mu) & 0 & 0 \\ \text{Sin}(\mu) & \text{Cos}(\mu) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} \text{Cos}(-\phi) & 0 & \text{Sin}(-\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sin}(-\phi) & 0 & \text{Cos}(-\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} 1 & 0 & 0 & r_\phi \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} \text{Cos}(\phi-\lambda) & 0 & \text{Sin}(\phi-\lambda) & 0 \\ 0 & 1 & 0 & 0 \\ -\text{Sin}(\phi-\lambda) & 0 & \text{Cos}(\phi-\lambda) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finally, multiplying the NEV Relative to ECEF Matrices, we obtain $M_{BCEF}^{NEV} =$

$$\begin{bmatrix} C(\mu)*[C(-\phi)*S(\phi-\lambda)+S(-\phi)*C(\phi-\lambda)] & -S(\mu) & C(\mu)*[S(-\phi)*S(\phi-\lambda)-C(-\phi)*C(\phi-\lambda)] & r_\phi*C(-\phi)C(\mu) \\ S(\mu)*[C(-\phi)*S(\phi-\lambda)+S(-\phi)*C(\phi-\lambda)] & C(\mu) & S(\mu)*[S(-\phi)*S(\phi-\lambda)-C(-\phi)*C(\phi-\lambda)] & r_\phi*C(-\phi)S(\mu) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4.2.1.2 Coordinate System Geometry - Entity Coordinate Reference

The entity (sometimes called vehicle or body) fixed reference frame is a coordinate system in which the origin is attached (fixed) to the entity at its center of mass (C). The next few paragraphs describe the ITB body-fixed reference system, the DIS entity-fixed reference system and the methodology required to translate between the two systems.

2.4.2.1.2.1 ITB Body Fixed Reference System

The definition of the coordinate system for the ITB body-fixed reference frame is such that the conventions for positive rotation of the platform through Roll, Pitch and Yaw correspond to positive rotations in a right-handed coordinate system. As a result, the following is true for the ITB body-fixed reference frame:

1. **X (U) Axis:** Positive direction points forward and aligned with the fuselage reference line.

2. **Y (V) Axis:** Perpendicular to the X axis with the positive direction pointing out the starboard (right) wing.

3. **Z (W) Axis:** Completes the right-handed coordinate system with the positive direction pointing downward.

4. **Origin:** Located at the center of mass of the body.

An illustration of the ITB body-fixed reference frame is provided in Figure 2.4.2.1.2.1-1.

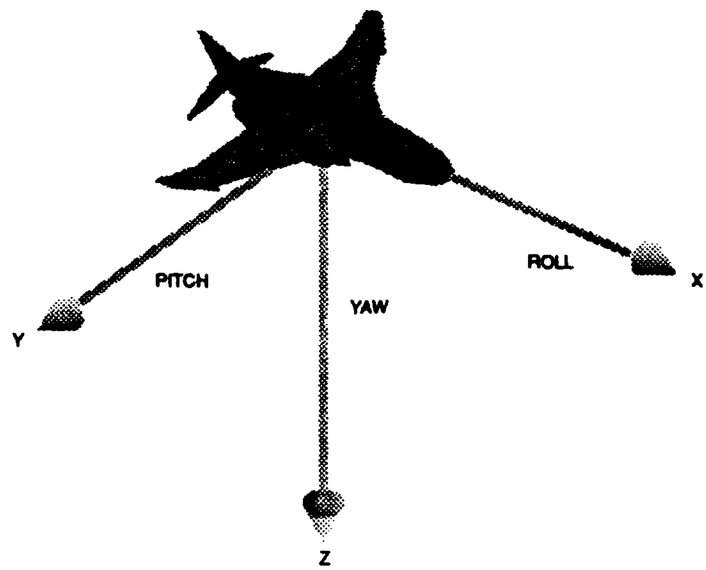

Figure 2.4.2.1.2.1-1 ITB Body-Fixed Coordinate System

2.4.2.1.2.2 DIS Entity Fixed Reference System

The definition of the coordinate system for the DIS entity-fixed reference frame is such that the conventions for positive rotation of the platform through Roll, Pitch and Yaw correspond to positive rotations in a right-handed coordinate system. As a result, the following is true for the DIS entity-fixed reference frame:

1. **X Axis:** Extends in the positive direction out the front of the entity.

2. **Y Axis:** Extends in the positive direction out the right side of the entity as viewed from above, facing in the positive direction of the X axis.

3. **Z Axis:** Completes the right-handed coordinate system. It extends in the positive direction downward.

4. **Origin:** Located at the center of the bounding volume of the entity.

An illustration of the DIS entity-fixed reference frame is provided in Figure 2.4.2.1.2.2-1.

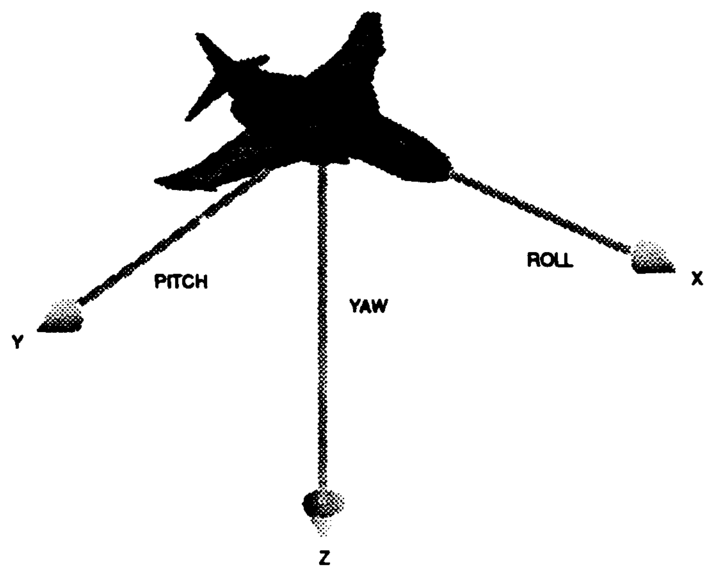

Figure 2.4.2.1.2.2-1 DIS Entity-Fixed Coordinate System

2.4.2.1.2.3 ITB-DIS Body-Fixed Coordinate Transformation Methodology

The ITB body-fixed and DIS entity-fixed reference systems are coincident (the same). No rotations or translations (transformations) are required.

2.4.2.1.3 Linear Velocity Vector

The DIS environment requires a linear velocity vector definition for the entity in question. The next few paragraphs describe the ITB linear velocity vector, the DIS linear velocity vector and the methodology required to translate between the two.

2.4.2.1.3.1 ITB Aircraft Linear Velocity Vector

The ITB supplies the aircraft linear velocity vector in Feet/Second as referenced to the ITB inertial (NEV) coordinate system. This vector is calculated in subroutine LINVEL of program AIRPLN and given the following symbols:

1. **X1DOTF:** Velocity of the aircraft in the X direction ("North") with respect to the NEV inertial coordinate system. Units are Feet/Second.

2. **Y1DOTF:** Velocity of the aircraft in the Y direction ("East") with respect to the NEV inertial coordinate system. Units are Feet/Second.

3. **Z1DOTF:** Velocity of the aircraft in the Z direction ("Down") with respect to the NEV inertial coordinate system. Units are Feet/Second.

2.4.2.1.3.2 DIS Entity Linear Velocity Vector

DIS requires the entity linear velocity vector to be in Meters/Second as referenced to the ECEF inertial coordinate system. The following applies for the DIS entity linear velocity vector:

1. **X Linear Velocity (X_{LV}):** Velocity of the entity in the X direction with respect to ECEF coordinates. Units are in Meters/Second.

2. **Y Linear Velocity (Y_{LV}):** Velocity of the entity in the Y direction with respect to ECEF coordinates. Units are in Meters/Second.

3. **Z Linear Velocity (Z_{LV}):** Velocity of the entity in the Z direction with respect to ECEF coordinates. Units are in Meters/Second.

2.4.2.1.3.3 ITB-DIS Linear Velocity Vector Transformation Methodology

The following paragraphs describes the methodology in order to translate the ITB Aircraft linear velocity vector to a DIS Entity Linear Velocity Vector:

1. Use the rotation portion (3x3) of the NEV relative to ECEF homogeneous matrix developed previously. We will denote this matrix M_{ij} where the subscripts ij denote the matrix row and column respectively:

$$M_{ij} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

2. Multiply this matrix by the ITB supplied aircraft linear velocity vector to obtain the aircraft linear velocity in ECEF coordinates with units of Feet/Second.

3. Finally, multiply the results of (2) by the Feet to Meters conversion factor of (0.3048). This produces the following result; answers are in Meters/Second:

$$X_{LV} = (X1DOTF*M_{11} + Y1DOTF*M_{12} + Z1DOTF*M_{13})*.3048$$
$$Y_{LV} = (X1DOTF*M_{21} + Y1DOTF*M_{22} + Z1DOTF*M_{23})*.3048$$
$$Z_{LV} = (X1DOTF*M_{31} + Y1DOTF*M_{32} + Z1DOTF*M_{33})*.3048$$

The following paragraphs describes the methodology in order to translate the DIS Entity linear velocity vectors into an ITB Aircraft linear velocity vector:

1. Use the rotation portion (3x3) of the NEV relative to ECEF homogeneous matrix developed previously. We will denote this matrix M_{ij} where the subscripts ij denote the matrix row and column respectively:

$$M_{ij} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}$$

2. Multiply the inverse of this matrix by the DIS supplied entity linear velocity vector to obtain the ITB aircraft linear velocity in NEV coordinates with units of Meters/Second.

3. Finally, multiply the results of (2) by the Meters to Feet conversion factor of (3.2808398). This produces the following result; answers are in Feet/Second:

$$X1DOTF = (X_{LV} * M_{11} + Y_{LV} * M_{21} + Z_{LV} * M_{31}) * 3.2808398$$

$$Y1DOTF = (X_{LV} * M_{12} + Y_{LV} * M_{22} + Z_{LV} * M_{32}) * 3.2808398$$

$$Z1DOTF = (X_{LV} * M_{13} + Y_{LV} * M_{23} + Z_{LV} * M_{33}) * 3.2808398$$

2.4.2.1.4 Linear Acceleration Vector

The DIS environment requires a linear acceleration vector definition for the entity in question. The next few paragraphs describe the ITB linear acceleration vector, the DIS linear acceleration vector and the methodology required to translate between the two.

2.4.2.1.4.1 ITB Aircraft Linear Acceleration Vector

The ITB supplies the aircraft linear velocity vector in Ft/S^2 as referenced to the ITB body-fixed coordinate system. This vector is calculated in subroutine ACCEL of program AIRPLN if the aircraft is airborne or in subroutine LACCEL of program AIRPLN if the aircraft is on or near the ground. The vector is given the following symbols:

1. **UDOTTF:** Acceleration of the aircraft in the X-Body direction with respect to the body-fixed coordinate system. Units are Ft/S^2.

2. **VDOTTF:** Acceleration of the aircraft in the Y-Body direction with respect to the body-fixed coordinate system. Units are Ft/S^2.

3. **WDOTTF:** Acceleration of the aircraft in the Z-Body direction with respect to the body-fixed coordinate system. Units are Ft/S^2.

2.4.2.1.4.2 DIS Entity Linear Acceleration Vector

DIS requires the entity linear acceleration vector be in M/S^2 as referenced to the DIS entity-fixed coordinate system. The following applies for the DIS entity linear acceleration vector:

1. **X Linear Acceleration (X_{LA}):** Acceleration of the entity in the X direction with respect to the entity coordinates. Units are in M/S^2.

2. **Y Linear Acceleration (Y_{LA}):** Acceleration of the entity in the X direction with respect to the entity coordinates. Units are in M/S^2.

3. **Z Linear Acceleration (Z_{LA}):** Acceleration of the entity in the X direction with respect to the entity coordinates. Units are in M/S^2.

2.4.2.1.4.3 ITB-DIS Linear Acceleration Vector Transformation Methodology

Due to the fact that both the ITB and DIS linear acceleration vectors are referenced to their respective "body-fixed" coordinate systems and the fact that both the ITB and DIS "body-fixed" coordinate systems are coincident (the same), no translations or rotations are necessary. Only a conversion between Ft/S^2 and M/S^2 is required. Therefore:

$$X_{LA} = (0.3048)*UDOTTF \quad (M/S^2)$$
$$Y_{LA} = (0.3048)*VDOTTF \quad (M/S^2)$$
$$Z_{LA} = (0.3048)*WDOTTF \quad (M/S^2)$$

and

$$UDOTTF=X_{LA} * (3.2808398) \quad (Ft/S^2)$$
$$VDOTTF=Y_{LA} * (3.2808398) \quad (Ft/S^2)$$
$$WDOTTF=Z_{LA} * (3.2808398) \quad (Ft/S^2)$$

2.4.2.1.5 Angular Velocity Vector

The DIS environment requires an angular velocity vector definition for the entity in question. The next few paragraphs describe the ITB angular velocity vector, the DIS angular velocity vector and the methodology required to translate between the two.

2.4.2.1.5.1 ITB Aircraft Angular Velocity Vector

The ITB supplies the aircraft angular velocity vector in Radians/Second as referenced to the body-fixed coordinate system. This vector is calculated in subroutine INTACC of program AIRPLN if the aircraft is airborne or in subroutine LINTAC of program AIRPLN if the aircraft is on or near the ground. The vector is given the following symbols:

1. **PBODYR:** Angular velocity of the aircraft about the X-Body axis (Roll Rate) with respect to the body-fixed coordinate system. Clockwise rotation is positive and the units are Radians/Second.

2. **QBODYR:** Angular velocity of the aircraft about the Y-Body axis (Pitch Rate) with respect to the body-fixed coordinate system. Clockwise rotation is positive and the units are Radians/Second.

3. **RBODYR:** Angular velocity of the aircraft about the Z-Body axis (Yaw Rate) with respect to the body-fixed coordinate system. Clockwise rotation is positive and the units are Radians/Second.

2.4.2.1.5.2 DIS Entity Angular Velocity Vector

DIS requires the entity angular velocity vector to be in Bams/Millisecond as referenced to the entity-fixed coordinate system. The following applies for the DIS entity angular velocity vector:

1. **X Angular Velocity (X_{AV}):** Angular velocity of the entity about the X-Entity axis (Roll Rate) with respect to the entity-fixed coordinate system. Counter-clockwise rotation is positive and the units are Bams/Millisecond.

2. **Y Angular Velocity (Y_{AV}):** Angular velocity of the entity about the Y-Entity axis (Pitch Rate) with respect to the entity-fixed coordinate system. Counter-clockwise rotation is positive and the units are Bams/Millisecond.

3. **Z Angular Velocity (Z_{AV}):** Angular velocity of the entity about the Z-Entity axis (Yaw Rate) with respect to the entity-fixed coordinate system. Counter-clockwise rotation is positive and the units are Bams/Millisecond.

2.4.2.1.5.3 ITB-DIS Angular Velocity Vector Transformation Methodology

Due to the fact that both the ITB and DIS angular acceleration vectors are referenced to their respective "body-fixed" coordinate systems and the fact that both the ITB and DIS "body-fixed" coordinate systems are coincident (the same), no translations or rotations are necessary. Only a conversion between Radians/Second and Bams/Millisecond as well as a conversion between Clockwise and Counter-Clockwise is required. If the Radians/Second to Bams/Millisecond conversion factor is given as 'C', then

$$X_{AV} = -(C)*PBODYR \quad \text{(Bams/Millisecond)}$$
$$Y_{AV} = -(C)*QBODYR \quad \text{(Bams/Millisecond)}$$
$$Z_{AV} = -(C)*RBODYR \quad \text{(Bams/Millisecond)}$$

and

$$PBODYR=-(1/C)*X_{AV} \quad \text{(Radians/Second)}$$
$$QBODYR=-(1/C)*Y_{AV} \quad \text{(Radians/Second)}$$
$$RBODYR=-(1/C)*Z_{AV} \quad \text{(Radians/Second)}$$

2.4.2.1.6 Location Vector

The DIS environment requires a location vector definition for the entity in question. The next few paragraphs describe the ITB location vector, the DIS location vector and the methodology required to translate between the two.

2.4.2.1.6.1 ITB Aircraft Location Vector

The ITB supplies the aircraft location vector in Feet as referenced from the center of the ITB inertial (NEV) coordinate system. This vector is calculated in subroutine INTVEL of program AIRPLN and given the following symbols:

1. **X1AIRF:** Location of the aircraft in the X direction ("North") with respect to the NEV inertial coordinate system. Units are Feet.

2. **Y1AIRF:** Location of the aircraft in the Y direction ("East") with respect to the NEV inertial coordinate system. Units are Feet.

3. **Z1AIRF:** Location of the aircraft in the Y direction ("Down") with respect to the NEV inertial coordinate system. Units are Feet.

2.4.2.1.6.2 DIS Entity Location Vector

DIS requires the entity location vector to be in Meters as referenced from the center of the ECEF inertial coordinate system. The following applies for the DIS entity location vector:

1. **X Location (X_{ECEF}):** Location of the entity in the X direction with respect to ECEF coordinates. Units are in Meters.

2. **Y Location (Y_{ECEF}):** Location of the entity in the Y direction with respect to ECEF coordinates. Units are in Meters.

3. **Z Location (Z_{ECEF}):** Location of the entity in the Z direction with respect to ECEF coordinates. Units are in Meters.

2.4.2.1.6.3 ITB-DIS Location Vector Transformation Methodology

The following paragraphs describes the methodology in order to translate the ITB Aircraft location vector to a DIS Entity location vector:

1. Use the NEV relative to ECEF homogeneous matrix developed previously. We will denote this matrix M_{ij} where the subscripts ij denote the matrix row and column respectively:

$$M_{Eij} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

2. Multiply this matrix by the ITB supplied aircraft location vector to obtain the aircraft location in ECEF coordinates with units of Feet:

$$\begin{bmatrix} X_{ECEF} \\ Y_{ECEF} \\ Z_{ECEF} \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & M_{Eij} & \\ & & \end{bmatrix} * \begin{bmatrix} X1AIRF \\ Y1AIRF \\ Z1AIRF \\ 1 \end{bmatrix}$$

3. Finally, multiply the results of (2) by the Feet to Meters conversion factor of (0.3048). This produces the following result; answers are in Meters:

$$X_{ECEF} = (X1AIRF * M_{11} + Y1AIRF * M_{12} + Z1AIRF * M_{13} + 1 * M_{14})*.3048$$
$$Y_{ECEF} = (X1AIRF * M_{21} + Y1AIRF * M_{22} + Z1AIRF * M_{23} + 1 * M_{24})*.3048$$
$$Z_{ECEF} = (X1AIRF * M_{31} + Y1AIRF * M_{32} + Z1AIRF * M_{33} + 1 * M_{34})*.3048$$

The following paragraphs describes the methodology in order to translate the DIS Entity location vectors into an ITB Aircraft location vector:

1. Use the NEV relative to ECEF homogeneous matrix developed previously. We will denote this matrix M_{ij} where the subscripts ij denote the matrix row and column respectively:

$$M_{Eij} = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & M_{22} & M_{23} & M_{24} \\ M_{31} & M_{32} & M_{33} & M_{34} \\ M_{41} & M_{42} & M_{43} & M_{44} \end{bmatrix}$$

2. Multiply the inverse (transpose) of this matrix by the DIS supplied entity location vector to obtain the ITB aircraft location in NEV coordinates with units of Meters:

$$\begin{bmatrix} X1AIRF \\ Y1AIRF \\ Z1AIRF \\ 1 \end{bmatrix} = \begin{bmatrix} & & & \\ & M^T_{Eij} & & \\ & & & \end{bmatrix} * \begin{bmatrix} X_{BCEF} \\ Y_{BCEF} \\ Z_{BCEF} \\ 1 \end{bmatrix}$$

3. Finally, multiply the results of (2) by the Meters to Feet conversion factor of (3.2808398). This produces the following result; answers are in Feet:

$$X1AIRF = (X_{BCEF} * M_{11} + Y_{BCEF} * M_{21} + Z_{BCEF} * M_{31} + 1 * M_{41}) * 3.2808398$$
$$Y1AIRF = (X_{BCEF} * M_{12} + Y_{BCEF} * M_{22} + Z_{BCEF} * M_{32} + 1 * M_{42}) * 3.2808398$$
$$Z1AIRF = (X_{BCEF} * M_{13} + Y_{BCEF} * M_{23} + Z_{BCEF} * M_{33} + 1 * M_{43}) * 3.2808398$$

2.4.2.1.7 Orientation Vector

The DIS environment requires a orientation vector definition for the entity in question. The next few paragraphs describe the ITB orientation vector, the DIS orientation vector and the methodology required to translate between the two.

2.4.2.1.7.1 ITB Aircraft Orientation Vector

The ITB supplies the aircraft orientation vector in Radians as referenced from the body-fixed coordinate system. This vector is calculated in subroutine INTVEL of program AIRPLN and given the following symbols:

1. **ACPSIR:** Euler Yaw angle. Units are Radians.

2. **THETAR:** Euler Pitch angle. Units are Radians.

3. **ACPHIR:** Euler Roll angle. Units are Radians.

2.4.2.1.7.2 DIS Entity Orientation Vector

DIS requires the entity orientation vector to be in Bams as referenced with respect to the entity-fixed coordinate system. The following applies for the DIS entity orientation vector:

1. **PSI_{DIS}:** Euler Yaw angle. Units are Bams.

2. **$THETA_{DIS}$:** Euler Pitch angle. Units are Bams.

3. **PHI_{DIS}:** Euler Roll angle. Units are Bams.

2.4.2.1.7.3 ITB-DIS Orientation Vector Transformation Methodology

Due to the fact that both the ITB and DIS Euler are referenced to their respective "body-fixed" coordinate systems and the fact that both the ITB and DIS "body-fixed" coordinate systems are coincident (the same), no translations or rotations are necessary. Only a conversion between Radians and Bams is required. If the Radians to Bams conversion factor is given as 'D', then

PSI_{DIS}=	(D)*ACPSIR	(Bams)
$THETA_{DIS}$=	(D)*THETAR	(Bams)
PHI_{DIS}=	(D)*ACPHIR	(Bams)

and

ACPSIR=	(1/D)*PSI_{DIS}	(Radians)
THETAR=	(1/D)*$THETA_{DIS}$	(Radians)
ACPHIR=	(1/D)*PHI_{DIS}	(Radians)

2.4.2.1.8 Dead Reckoning

The DIS environment requires that a specific method of position/orientation estimation (called Dead Reckoning) be employed in order to limit the rate at which Entity State PDUs are issued. Each host computer maintains a high fidelity model of itself (representing its actual position) and a lower fidelity, dead reckoned model of itself. Certain thresholds are established as criteria for determining if the entities actual position/orientation has varied from its dead reckoned position/orientation. When the entities actual position/orientation has varied form the dead reckoned position/orientation by more than the threshold, then the entity issues an Entity State PDU to communicate to the other host computers its actual position. The entity also uses the same information communicated to other host computers to update its own dead reckoning model of itself.

Each host computer also maintains a dead reckoned model of the position/orientation of entities that are of interest as specified by the dead reckoning model in use. The dead reckoned position/ orientation of the other entities are used to display their position/orientation in the host entities visual or sensor displays. When the host entity receives a new update from one of the entities it is dead reckoning, it corrects its dead reckoned model and bases its estimations on the most recent position/orientation velocity and acceleration information.

The dead reckoning algorithms required for the DIS-ITB interface will be those supplied in the DIS Military Standard document.

2.4.2.2 Entity Database

Figure 2.4-2 illustrates the Entity Database, which provides input to the PDU processor. The purpose of the Entity Database is to supply detailed simulation information about *entity types*. The Entity Database will be a text file comprising records that include information about each type of entity that will be encountered in a scenario. The key to each record will consist the following fields from an Entity Type in an Entity State PDU.

- Entity kind
- Country
- Category
- Subcategory
- Specific

For example, the key for an American F-15E aircraft is as follows:

- Entity kind = 1
- Country = 168

- Category = 1
- Subcategory = 7
- Specific = 0 (not applicable)

The data in each record consists of information to support realistic representation of the entity in the ITB. The following data will be part of each Entity Database record.

- BBN Display ID
- Relative size of entity, actual values TBD
- Dimensions of entity, actual values TBD pending full definition of collision determination algorithm

During ITB initialization, the Entity Database will be read into ITB random access memory for efficient access during simulation. When an Entity State PDU is received for an entity that has not yet been encountered, its record will be accessed and information placed in its Entity Data Structure record. There should be an Entity Database record for each entity type in the scenario. There will be provisions for specifying generic information so that unknown entity types will be rudimentarily represented.

2.4.2.3 Use of Terrain and Cultural Databases

Project 2851 has been making rapid progress in standardizing terrain and cultural database interpretation. Two military standards have been produced. The first, called the Generic Transformed Data Base (GTDB) is intended for release of original information. The second, called Standard Simulator Data Base (SSDB) Interchange Format (SIF), is meant to support the exchange of enhanced/customized databases between sites and simulators. SIF seems to be becoming the de facto standard for terrain representation in image generators. SIF uses ASCII format, which greatly simplifies its interpretation, and is easier to use (based on preliminary comments) than GTDB. However, the SIF standard is being revised, with release expected in early 1993.

At the Interservice/Industry Training Systems Conference (I/ITSC) in November, 1992, there will be an interoperability demonstration using DIS. The purpose of the demonstration, which is an initiative of the Defense Modeling and Simulation Office (DMSO), is to encourage the adoption of DIS for the network protocol and SIF for the database interchange format. DMSO intends to seed the development of a technical support infrastructure in these areas. This is being accomplished through the conversion of the Hunter-Liggett database into SIF format. This database will be provided to all participants to convert into their internal image generator format for use in the demonstration. Use of this database is being strongly encouraged.

SIF capability for the ITB is highly recommended. SIF data would be required for the OTW/FLIR display, the HUD, and the ITB internal processing. Use of SIF data would probably require modification of the internal processing software for the OTW/FLIR and HUD. These displays are produced by BBN. BBN has been following Project 2851 work and attending conferences, and should be aware of the work involved. SIF terrain data is gridded in a similar manner to what is currently used by the ITB software. It has not been possible to determine the level of effort required to use SIF data, since the document we used did not have the detailed formats or examples. Experiences and lessons learned from the I/ITSC demonstration participants could be applied directly to Phase II efforts.

Appendix A

ACRONYMS AND ABBREVIATIONS

ADSS	Availability Demonstration Simulation Support
AFB	Air Force Base
AFIT	Air Force Institute of Technology
AKA	Also Known As
ASCII	American Standard Code for Information Interchange
BAM	Binary Angular Measurement
BBN	Bolt Beranek and Newman
BIU	Bus Interface Unit
CAS	Close Air Support
CDRL	Contract Data Requirements List
CNI	Communication, Navigation, and Identification
DIS	Distributed Interactive Simulation
DMA	Direct Memory Access
DMSO	Defense Modeling and Simulation Office
E	East
ECEF	Earth-Centered Earth-Fixed
FLIR	Forward Looking Infrared
GENASIS	Generalized Avionics Simulation/Integration System
GPS	Global Positioning System
GTDB	Generic Transformed Data Base
HUD	Head's-Up Display
Hz	Hertz
I/ITSC	Interservice/Industry Training Systems Conference
I/O	Input/Output
IBM	International Business Machines
ID	Identifier
IEEE	Institute of Electrical and Electronics Engineers
IESS	Integrated Electromagnetic Simulator System
IST	Institute for Simulation and Training
ITB	Integrated Test Bed
LAN	Local Area Network
MBT	Multiplex Bus Terminal
MSS	Model Simulation System
N/A	Not Applicable

NEV	North-East-Vertical
OSI	Open System Interconnection
OTW	Out-the-Window
PC	Personal Computer
PDU	Protocol Data Unit
RF	Radio Frequency
SBIR	Small Business Innovative Research
SIF	SSDB Interchange Format
SIMNET	Simulator Network
SSDB	Standard Simulator Data Base
TBD	To Be Defined
UDP/IP	User Datagram Protocol/Internet Protocol
UTC	Universal Coordinated Time
WGS-1984	World Geodetic System 1984 Survey
WL	Wright Laboratory

Appendix B

RELATED DOCUMENTS

"12th Interservice/Industry Training Systems Conference 1990: SIMNET Fighter Aircraft Application", Armstrong Laboratory Aircrew Training Research Division, Document No. AL-TR-1992-0020, March 1992.

"BBN GT100 CIG to Simulation Host Interface Manual," BBN Systems and Technologies, Document No. 8912, July 1990.

"Computer Program Development Specification for the Models Simulation System (MSS) (STE0001), Part II," WL/AAAS-2, Document No. SSTE11101, 18 July 1990.

"Computer Program Development Specification for the Models Simulation System (MSS), Part I," WL/AAAS-2, Document No. SSTE11101, 7 March 1989.

"GT100 Interface and Applications Development," BBN Systems and Technologies, undated.

"Interface Control Document For Generic Trainer/Network Interface Unit," LORAL Defense Systems Akron, April 1992 (Preliminary).

"Military Standard, Standard Simulator Data Base (SSDB) Interchange Format (SIF) for High Detail Input/Output (SIF/HDI) and Distributed Processing (SIF/DP)," AMSC/NA, December 1990 (Draft).

"Network Interface Unit (NIU) Detailed Design Specification," BBN Systems and Technologies, Report No. 7622, June 1991.

"PAVE PILLAR In-House Technical Report Volume I - Demonstration I and IA," WL/AAAS-2, July 1990.

"Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation," Institute for Simulation and Training, 30 October 1991.

"Rationale Document, Entity Information and Entity Interaction in a Distributed Interactive Simulation," Institute for Simulation and Training, January 1992.

"Segment Specification for the Real-Time Support System (RSS) (STE0001),' WL/AAAS-2, Document No. SSTE10101, 28 July 1990.

"Simulator Networking Phase I Technical and Management Work Plan," Amherst Systems Incorporated, Document No. 611-9420001 rev. A, 22 May 1992.

"Simulator Networking Phase I, Contractor's Progress, Status and Management Plan, February 1, 1992 Through April 30, 1992," Amherst Systems Incorporated, Document No. 611-9160001, 29 May 1992.

"Summary of SIMNET 6.6 Implementation and Extensions for TRUE," LORAL Defense Systems-Akron, 14 April 1992 (Draft).

"System Specification for the Integrated Test Bed (ITB) Facility," WL/AAAS-2, Document No. STZ0001, 23 May 1989.

"The SIMNET Network and Protocols", BBN Systems and Technologies, Report No. 7627, June 1991.